Chapter Seven: Contents

(Emissions Estimators – 09 April 2002 – LA-UR-00-1725 – TRANSIMS 3.0)

Chapter Seven: Figures

Chapter Seven—Emissions Estimator

The TRANSIMS Emissions Estimators module measures two types of emissions:

- Tailpipe emissions, and
- Evaporative emissions.

1. Tailpipe Emissions

1.1 Overview

The TRANSIMS Tailpipe Emissions Estimator translates traveler behavior into emissions (of nitrogen oxides, hydrocarbons, carbon monoxide, and carbon dioxide) and energy consumption. The calculated emissions can then be used with an air-shed model to calculate pollutant concentrations for a metropolitan area.

The present version estimates tailpipe emissions from light-duty vehicles (LDVs) and heavy-duty vehicles (HDVs).

With regard to off-cycle conditions, very high emissions take place at high-power demands. The phrase "off-cycle" refers to conditions outside those that occur in the federal test procedure[1]. Emissions in this context are very sensitive to the precise acceleration that takes place at a specific speed.

Fig. 1 summarizes the information flow of the Tailpipe Emissions Estimator. This module requires information regarding

- the fleet composition developed from the Population Synthesizer, and
- traffic patterns produced by the Traffic Microsimulator.

[1] FPT (1989). Code of Federal Regulations. Title 40, Parts 86-99 (portion of CFR that contains the Federal Test Procedure), Office of the Federal Register.

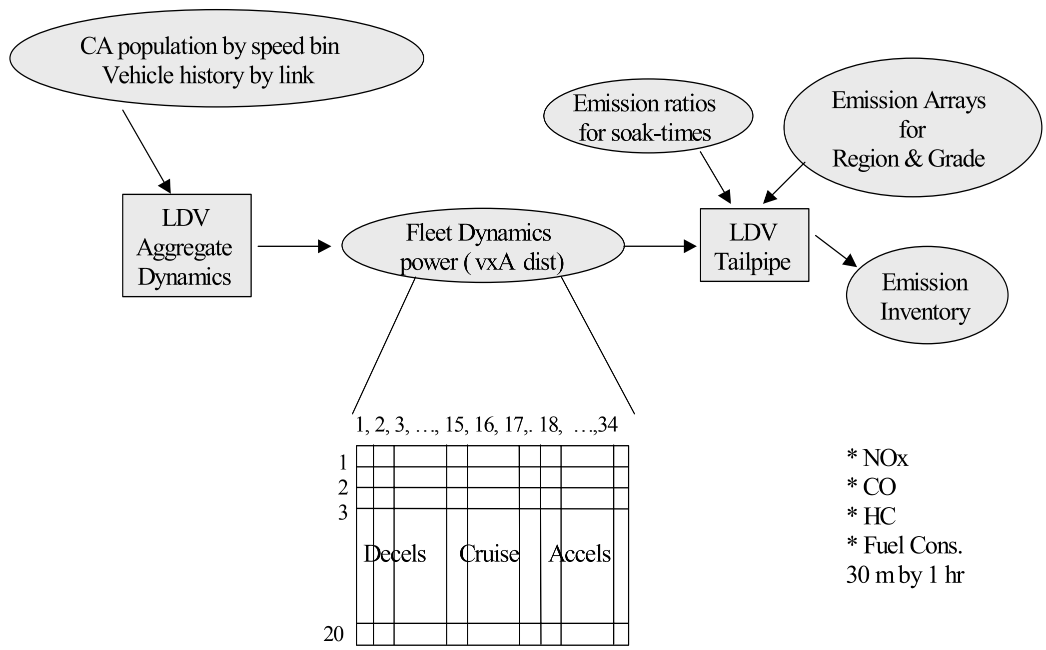

Fig. 1. This flowchart shows the process used to produce an inventory of a gridded LDV emission.

The system's output consists of emission estimates on 30-meter segments for each one-hour period simulated. Fuel economy and CO_2 emissions are also estimated.

The emission inventory is designed to be used with the Environmental Protection Agency's (EPA's) MODELS-3 to produce three-dimensional hourly gridded emissions over the metropolitan area.[2] MODELS-3 is a modeling system that includes an emissions estimation model, a meteorology model, and an air-chemistry and dispersion model.

1.2 Light-Duty Vehicle (LDV) Tailpipe Emissions Estimator Submodule

The LDV Tailpipe Emissions Estimator submodule treats tailpipe emissions from cars, small trucks, and sport-utility vehicles. The submodule covers a number of scenarios, such as the following:

- malfunctioning vehicles,
- emissions from cold starts,
- emissions from warm starts in which the engine is still warm but the catalyst is cold,

[2] Novak, J.H., R.L. Dennis, D.W. Byun, J.E. Pleim, K.J. Galluppi, C.J. Coats, S. Chall, and M.A. Vouk, "EPA Third-Generation Air Quality Modeling System: Volume 1 Concept," EPA600/R95/082, 1995, US Environmental Protection Agency, Research Triangle Park, NC 27711.

- emissions from off-cycle conditions that render the pollution controls inefficient, and
- normal driving.

There are three major sets of information that must be developed:

1) What is the fleet composition?
2) What is the fleet status?
3) What is the fleet doing?

Once these questions are answered, the LDV Tailpipe Emissions Estimator can produce the emissions.

1.2.1 Fleet Composition

Fleet composition is developed from vehicle registration data, inspection and maintenance testing, or data developed for the Environmental Protection Agency's (EPA's) mobile model runs. Barth and his colleagues[3] have developed techniques to take registration data and produce vehicle populations in each of 23 categories. The categories include factors such as

- low or high engine-to-weight ratio,
- car or truck,
- mileage above or below 50,000,
- type of catalyst (2-way or 3-way),
- carbureted or fuel-injected, and
- high or normal emitting.

The Emissions Estimator produces an improved estimate of the proportion of high-emitting vehicles in cases in which there is a sophisticated inspection and maintenance program that tests vehicles on a dynamometer.

In this model's current version, the vehicle distribution is used to compute the relationship among emissions and speeds and power (more precisely, velocity acceleration product) for a composite vehicle representative of the fleet in Southern California. The array parameter files embody these relationships and are used in the LDV Tailpipe Emissions Estimator. For other locales, including Portland, mobile model input is used to calculate fleet distribution. The distribution is used to calculate array parameter files specific to the region being modeled.

[3] Barth, M., T. Younglove, T. Wenzel, G. Scora, F. An, M. Ross, and J. Norbeck (1997), "Analysis of Modal Emissions for a Diverse in-use Vehicle Fleet." Transportation Research Record, No. 1587, Transportation Research Board, National Academy of Science, pp 73-84.

1.2.2 Fleet Status

Fleet status is developed from the usage pattern of vehicles traversing a given link. The Traffic Microsimulator keeps track of when and where the vehicles have been operating. Cold engines burn fuel-rich until the engine has burned enough fuel to bring the engine temperature up to normal. Similarly, the catalyst efficiency is reduced until enough fuel has been burned to bring the catalyst up to normal operating ranges. Within a given vehicle category, power demand is the principal determinant of fuel consumption.

To represent the distribution of vehicles in various warm-up stages, we gather the vehicles entering the link into groups based on:

- their integrated product of speed and acceleration since the last start of the engine, and
- the soak time between the current operation and the end of the last trip.

In this code's version, there are three soak times, based on a 20-minute, one hour, and five-hour time between starts. Shorter soak times (down to 10 minutes) would reduce the highest ratios of cold-to-warm emissions by approximately 10% for hydrocarbons. For NO_x, shorter soak times have a greater effect, but the difference between cold and warm emissions is much less.

There are seven groupings based on velocity-acceleration product, and there is an additional grouping for engines that have been fully warmed-up—for a total of eight groups. The integrated velocity-acceleration product is in units of cells-squared per second-squared; a cell is 7.5 meters.

For each group, we assign a multiplier for each parameter: hydrocarbons, carbon monoxide, nitrogen oxides, and fuel consumption. The multiplier represents an emissions ratio for vehicles beginning a link in the group to the emissions of a vehicle (with the same driving pattern) with a fully warmed-up engine and catalyst.

A data statement for variables `hcr`, `cor`, `xnoxr`, and `fcr` provides the ratios for hydrocarbons, carbon monoxide, oxides of nitrogen, and fuel consumption, respectively. In each case, the first element of the array gives a soak time of 20 minutes and the lowest integrated velocity-acceleration product, whereas the last element is "one" because it represents the ratio of emissions from warmed-up vehicles to emissions of warmed-up vehicles.

To obtain these groupings and values, we took several actual trajectories in which vehicles accelerated from a near stop (less than 5-mph) at a traffic controller signal and achieved typical speeds on an arterial. The original trajectories were approximately 30 seconds long and covered approximately one-quarter mile.

The ends of these trajectories were replaced with a short deceleration to the initial speeds, then the trajectories were repeated. In this way, we obtained several trajectories, with 10 cycles of accelerating from a near stop and achieving speeds and decelerating.

The stops were selected to be approximately one-quarter mile apart. These trajectories were analyzed with the Comprehensive Modal Emission Model[4] (CMEM) to obtain emissions for a soak time of 60 minutes. Fig. 2 shows the hydrocarbon emissions for cold engine relative to a fully warmed-up engine and catalyst.

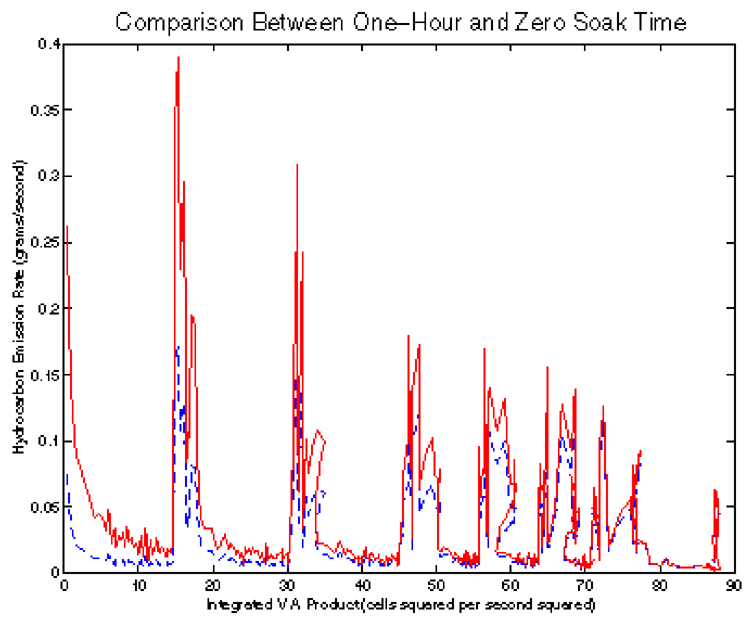

Fig. 2. Emissions from a vehicle with one hour of engine off before the trajectory compared with those from a vehicle with a warm engine and catalyst.

These results were used to construct ratios for each cycle by integrating the emissions curves from the start of one peak to the beginning of the next peak. For example, by integrating from 0 to 15, the cold engine has approximately three times the emissions of the warm engine. After several stop-start cycles, the ratios are lower for the other pollutants and the all-approach one. Fig. 3 reports the calculated ratios for various pollutants.

[4] Barth, M., T. Younglove, T. Wenzel, G. Scora, F. An, M. Ross, and J. Norbeck (1997), "Analysis of Modal Emissions for a Diverse in-use Vehicle Fleet." Transportation Research Record, No. 1587, Transportation Research Board, National Academy of Science, pp 73-84.

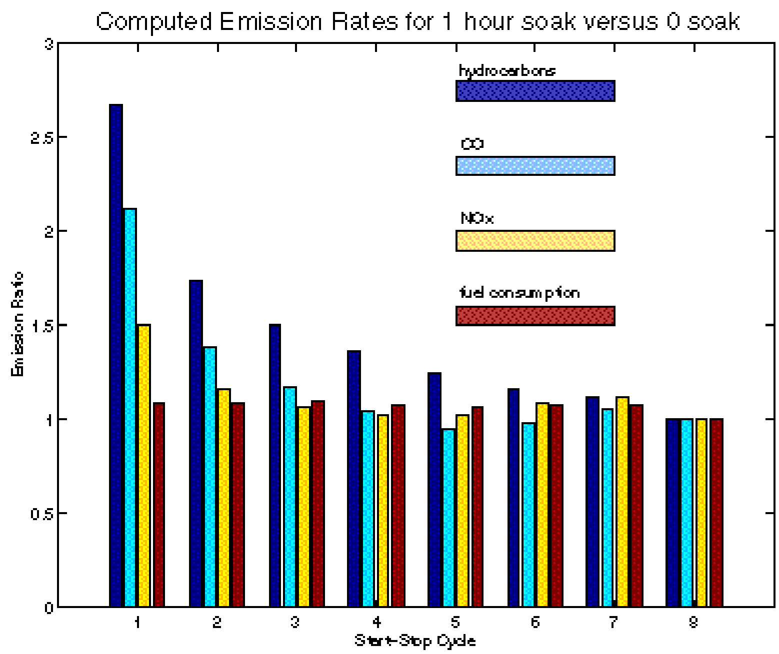

Fig. 3. Ratio of cold engine emissions to warmed-up vehicle emissions as a function of the number of start-stop cycles since the engine was turned on.

The ratios for the eight combinations of integrated velocity-accelerator product are used to compute a multiplier that converts emissions from warmed-up vehicles into emissions for the fleet on each link. The same procedure was used for soak times of 20 minutes and 5 hours.

1.2.3 Fleet Dynamics

To develop fleet dynamics information, we use microsimulation cellular automata (CA) populations grouped into each possible microsimulation speed (called speed bins) and grouped by 30-meter segments. One of the major challenges of appropriately modeling emissions is to account for the effects of different power demands by different drivers.

With LDVs, there is a range of accelerations available to the driver. The driver who favors harder accelerations may put the vehicle into an "enrichment" mode. During enrichment conditions, the vehicle's fuel controls switch to a fuel-rich situation that produces high emissions from the engine. Because the catalyst is starved for oxygen, it does not significantly reduce the emissions. Although this fuel-control logic protects the catalyst from getting too hot, it enormously increases the vehicle's emissions.

Because the federal test procedure[5] was designed at a time when dynamometers could not make measurements during high-power circumstances, most vehicles during the test spent little, if any, time in the enrichment mode. Thus, vehicles can pass EPA emission requirements even though they may have very high emissions on the highway.

Although enrichment episodes take place in a relatively small proportion of the time, the emissions are so much higher than normal that they can produce a significant fraction of the total emissions. Although emissions of pollutants such as NO_X are relatively unaffected by enrichment conditions, they nevertheless are quite sensitive to power levels.

The upshot of all this is that it is not enough to describe the power levels demanded by a typical driver. The actual range of driving behavior must be represented. Because enrichment is expected to take place only one to a few percent of the time, the range of power levels must be described for the less than one percent of the people who drive most aggressively.

1.2.3.1 Driving on Hills

For driving on hills, enrichment becomes much more frequent, with some cars going into enrichment while merely maintaining speed on uphill portions of major highways. In addition, highway speeds significantly higher than those encountered in the federal test procedure can also produce enrichment conditions. There are two options designed to address this challenge.

Option One Construct a fast, accurate microsimulation that describes traffic and power demands in great detail.

Option Two Supplement a fast microsimulation that describes traffic properly with empirical information on power demands.

One of the major difficulties with Option One is finding adequate information that describes the range of driving behavior in specific circumstances. There is much more information on traffic than there is on individual driving speeds and accelerations. Because of this, we have selected Option Two.

1.2.3.2 Traffic Microsimulator

We have developed a Traffic Microsimulator module that describes traffic accurately and efficiently. From the microsimulation, we know the context in which driving occurs, and we have developed a system—the LDV Aggregate Dynamics submodule—to place empirical information into context.

[5] FTP (1989) Code of Federal Regulations. Title 40. Parts 86-99 (portion of CFR that contains the Federal Test Procedure), Office of the Federal Register.

The Traffic Microsimulator provides vehicle populations by discrete speed bins in 30-meter segments from the start of each link in each direction. The first step is to develop a continuous distribution of speeds. We interpret the CA populations in a given segment as follows:

- The integral of a continuous distribution described by the mean value and a slope term proportional to the difference between a given speed in the speed bin and the center of the speed bin.

We have two constants to determine and two relationships to use for each speed bin and each segment:

1) the CA speed populations, which provide one equation for each speed bin; and

2) the continuity requirements between speed bins and the restrictions that the density must go to zero at the top of the highest speed bin.

The next question is as follows: How is the distribution of accelerations determined for a given speed in a specific segment? We use two sets of empirical data to help us solve this problem. First, during EPA's three-city studies[6], many vehicles were fitted with a data logger that recorded times and speeds throughout the vehicle's travels for a significant period.

These data were examined to determine the frequency distribution of accelerations for a given speed. More specifically, we looked for the cumulative frequency of positive accelerations. Fig. 4 shows the cumulative frequency of vehicles with positive accelerations traveling faster than one cell per second (7.5 m/s); these vehicles also have positive accelerations plotted against the product of acceleration and speed.

[6] USEPA (1993), "Federal Test Procedure Review Project: Preliminary Technical Report," EPA 420-R-93-007, 1993, Office of Air and Radiation, Washington, DC.

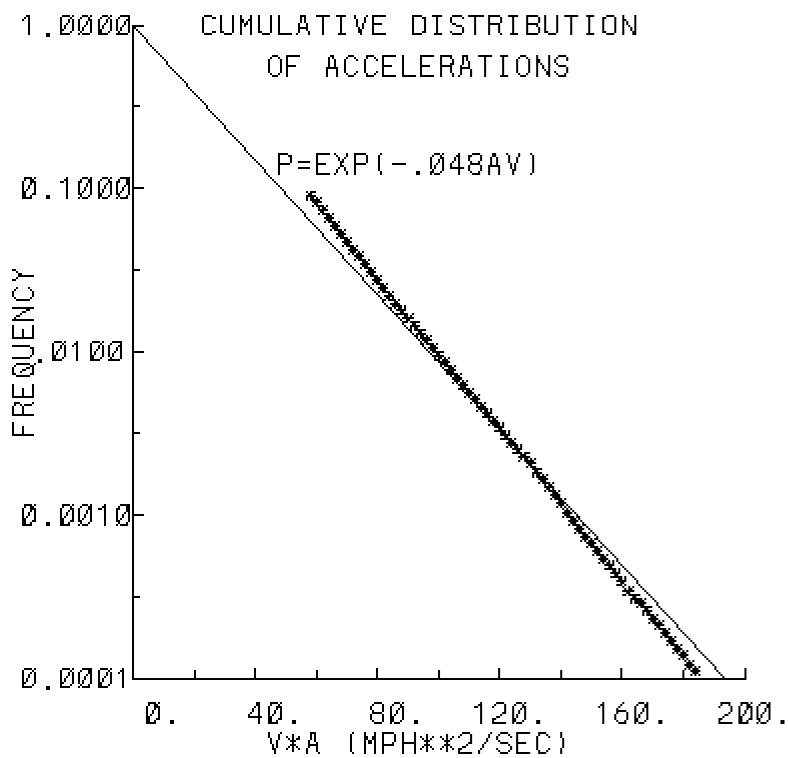

Fig. 4. The cumulative distribution of accelerations from the EPA's three-city studies.

1.2.3.3 Power Demands

For higher power levels, the frequency of a given power level falls off exponentially with power. Similar plots can be made for speeds less than one cell per second and for decelerations. In the case of decelerations, the frequency falls off exponentially with the velocity-deceleration product.

These relationships form one of the empirical underpinnings of our approach. We consider all accelerations in one of three groups:

- high power,
- insignificant power, and
- low power.

High-power demands are defined by velocity-accelerations greater than 50-mph squared per second, which corresponds with the 10% point on the cumulative distribution. Conversely, low-power demands are defined by velocity-acceleration products less than –50-mph squared per second.

In our approach, we begin by estimating the number of vehicles that are demanding high power. We then select 15 different power levels to represent different levels of aggressiveness. These levels are selected from the curve found in Fig. 4; they have equal spacing in power and cover a range from a cumulative frequency of 0.1 to 0.0045. The total vehicle population demanding high power from a given speed is then distributed over the 15 power (or, equivalently, acceleration) levels (in accordance with Fig. 4).

There are two circumstances in which the basic frequency power curve is significantly different from that of Fig. 4:

- freeway on-ramp driving, and
- driving at the end of signalized links

In freeway on-ramp driving, the frequency falls off much slower with power. In this case, we use 90-mph squared per second as the definition of a high-power demand and –90-mph squared per second as the definition of a low-power demand. For the ends of signalized links, the curve for frequency versus the deceleration-speed product falls off much slower with negative power than it does for other circumstances.

The next issue involved is determining the fraction of vehicles that have a high-power demand in a given context. We expect that a small fraction of vehicles will be undergoing a hard acceleration, even though most have reached their desired driving speed. We can estimate this low level by looking at uncongested freeways.

We expect that a somewhat larger fraction of the vehicles will undergo hard accelerations if they are driving on a moderately congested freeway in which they are forced to decelerate (to avoid hitting other cars) then accelerate (to regain their speed). If we looked at the speeds accumulated over time of such vehicles, we would find average speeds less than the desired maximum and a range of speeds from low ones to the desired maximum.

Under these circumstances, the standard deviation of the speeds would be relatively large. In this case, we could estimate the frequency of hard accelerations by comparing moderately congested freeways with uncongested freeways.

We also would expect significant hard accelerations to take place when many of the vehicles are accelerating, and as a result, the average speed increases as they proceed along the link. For example, if we look at vehicles leaving an intersection with a stoplight, there will be vehicles going through with a green light and those accelerating from a stop caused by a red light.

To characterize the hard accelerations that result from a stoplight, the Emissions Estimator uses the product of acceleration and speed. Because acceleration is expressed as the velocity times the rate of change of velocity with distance, we require the square of the velocity times its rate of change with distance.

This term is appropriately weighted by the number of vehicles in a spatial segment that have speed v. In this context, we can estimate the fraction of accelerations that are hard by examining an arterial and looking at the vehicles leaving a stop light.

Because of the importance of appropriately treating vehicles leaving an intersection, we use regression relationships to estimate the vehicle populations by speed bin in the 30-meter segment that precedes the link; we also perform emission estimates for it. In addition, we average the speed gradients over the three adjacent segments (including the added segment that represents the intersection for the first segment of the link). Averaging over three segments compensates for the faster average accelerations found in the cellular automata.

1.2.3.4 Empirical Data

The Emissions Estimator required empirical data that cover driving on uncongested freeways, moderately congested freeways, and arterials. The second empirical underpinning for our submodule comes from data collected by the California Air Resources Board[7].

The board's contractor used a laser range-finder to follow cars and record their speeds under a variety of circumstances. They produced seven sets of individual car trajectories organized by congestion on freeways. They also produced three sets of trajectories that they characterized as slow, medium, and fast arterials.

To construct a two-parameter fit, the data from the fastest freeway, a moderately congested freeway, and the fast arterial were used. In the case of the arterials, the trajectories were selected so that they started (within one-second precision) from a signalized intersection. We selected transformations of the acceleration-velocity product and the standard deviation of speed to the fraction of high-power driving such that when we used the high-power fractions, we would get the same power in high-power driving that we would get by using the original trajectories.

We used a similar approach to estimate the fraction of vehicles in low-power driving. The difference between the two gives the fraction of vehicles in intermediate-power driving. By integrating the high- and low-power curves, we get the total power in low- and high-power driving. The total power in all driving can be estimated from the integrated velocity-acceleration product.

The power in intermediate-power driving is obtained by subtracting the power in high- and low-power driving from the total power. This permits us to calculate the average power in intermediate-power driving because we know that (1) the total power in intermediate power, and (2) the number of vehicles demanding intermediate power.

1.2.3.5 On-ramps

In the case of on-ramps, an investigator[8] collected speed and acceleration distributions on freeway on-ramps in California. We used these data to estimate trajectories for one of the on-ramps. We used a similar approach to the one described above to obtain the calibration constants for freeway on-ramps. The on-ramp was a 2% grade so we added in

[7] Effa, Robert C. and Lawrence C. Larson, "Development of Real-World Driving Cycles for Estimating Facility-Specific Emissions from Light-Duty Vehicles," California Environmental Protection Agency – Air Resources Board – presented at the Air and Waste Management Association Specialty Conference on Emission Inventory 1993, Pasadena, California.

[8] Sullivan, Edward C. and Alypios Chatzijoanou (1993), "Vehicle Speeds and Accelerations Along On-Ramps: Input to Determine the Emissions Effects of Ramp Metering – Final Report," prepared for Caltrans Office of Traffic Improvement, Sacramento, California, California's Polytechnic State University, San Luis Obispo, California.

the power required to drive at each speed up a 2% grade before we estimated the calibration constants.

In a similar fashion, we reorganized our arterial trajectories so that all vehicles approached a signalized intersection at the same spatial position. We used the fast arterial trajectories to obtain the calibration constants for decelerations at the end of a link. We also tested the formulation on the other arterials and on-ramps.

1.3 LDV Aggregate Dynamics Submodule

The LDV Aggregate Dynamics submodule carries out the continuous fit and computes the fraction of vehicles in each driving mode and the average power for the vehicles in the intermediate-power mode. When we compared the emissions from the other freeways and the two other arterials from the original trajectories with those obtained from our LDV Aggregate Dynamics submodule, we obtained good agreement.

The output of the LDV Aggregate Dynamics submodule consists of the number of vehicles in each 30-meter segment in each of 20 4-mph speed bins undergoing one of 34 different levels of acceleration-speed power.

The speed-acceleration levels include

- 18 high-power levels,
- one insignificant level, and
- 15 low-power levels.

This two-dimensional array is produced for each 30-meter segment of each link for each one-hour period. The power of the intermediate level is also calculated.

1.4 CMEM Model

TRANSIMS uses the CMEM model developed by Matt Barth[9] and his colleagues at the University of California at Riverside and the University of Michigan. Barth and his co-investigators were contracted by the National Cooperative Highway Research Program to develop an improved modal emission model for LDVs.

They carried out extensive tests on more than 300 vehicles selected to represent the major types of emitters in the existing LDV fleet. They also worked with other data to help draw associations between the tested vehicles and the fleet at large.

CMEM computes the tractive power by taking account of

- engine friction losses,
- rolling resistance,

[9] Barth, M., T. Younglove, T. Wenzel, G. Scora, F. An, M. Ross, and J. Norbeck (1997), "Analysis of Modal Emissions for a Diverse in-use Vehicle Fleet." Transportation Research Record, No. 1587, Transportation Research Board, National Academy of Science, pp 73-84.

- wind resistance,
- changes in kinetic energy,
- changes in potential energy, and
- the power necessary to drive accessories such as air conditioning.

It also estimates drivetrain efficiency. With the engine power known, CMEM calculates the rate of fuel consumption and engine-out emissions. It treats enrichment, enleanment, and stoichiometric operations, as well as cold-start operation.

Once the engine-out emissions are calculated, catalyst pass fractions are used to calculate tailpipe emissions. This approach uses a composite vehicle to represent vehicles in the same class.

A regression approach was used to define the parameters required by the model. The vehicles were all tested over cycles involving (1) very high power demands, and (2) a variety of driving patterns.

The Emissions Estimator has in place composite vehicles that represent normal emitting vehicles categorized by technology, low and high power-to-weight ratios, and mileages above or below 50,000. The technology categories are as follows:

- no catalyst,
- two-way catalyst,
- three-way catalyst with carburetor,
- three-way catalyst with fuel injection, and
- Tier 1.

Only the last two technologies are broken into mileage or power-to-weight ratio groupings. There are high-emitting composite vehicles for technologies 3 through 5, but they are not further subdivided into power-to-weight ratios or mileage groupings.

There are composite vehicles representing normal-emitting trucks with the following model year categories:

- pre-1979,
- 1979 to 1983,
- 1984 to 1987,
- 1988 to 1993, and
- 1994 and newer.

In the age groupings pre-1979, 1979 to 1983, and 1984 to 1987, there is only one single composite vehicle for each grouping.

For the age grouping 1988 to 1993, there are separate composite vehicles for trucks above 3,750 pounds loaded vehicle weight and below 3,750 pounds loaded weight.

For 1994 and newer vehicles, there are two composite vehicles representing vehicles with weights less than 5,750 and vehicles with weights greater than 5,750 but less than 8,500 pounds.

There are also composite vehicles representing high-emitting trucks for model years 1984 to 1987, 1988 to 1993, and 1994 and newer.

In the high-emitting category, there are no breakdowns by vehicle weight.

The relationships that CMEM produces between speed and acceleration (actually, speed times acceleration) are embodied in the array parameter files . These files give the composite vehicle emissions for 4-mph speed bins and 20-mph squared per second squared power bins.

The arrays were developed by using CMEM for four-second trajectories ending in the specified speed bin and power associated with the selected power bin. The arrays were calculated for constant power accelerations (if possible). For high power and low speeds, it is not possible to have constant power accelerations without starting speeds less than zero. Consequently, starting speeds were selected to give the best approximation to constant power with the constraint that the starting speed was greater than 0.1-mph.

In addition, we compute arrays that give the difference in emissions between constant power trajectories and those with the same speed and power, but with a step change in power over the previous second.

In the case where the power is high (above the 50-mile-per-hour squared per second cutoff), the jump is between zero power and the desired power. In this instance, the code estimates the fractional change in power between the second of interest and the preceding second. The emissions are obtained by multiplying the fractional power change times the emission difference for the given speed and power, then by adding the result to the emissions at constant power. This approach enables us to address history effects associated with large increases in power. When the power is low, the jump is from a higher power down to zero power with the magnitude of the jump based on the difference between the current power and that at the previous second. This approach is used to capture the effects of rapid decelerations.

The basic relationships that give the probability of high or low power driving in terms of either the gradient in speed or the standard deviation of speed were developed for zero grade driving. Consequently, the power associated with a specific kind of driving is adjusted for the power required to drive the vehicle up or down the slope before it is used in the emission arrays. The emission arrays are calculated for slopes of 0, 2, 4, and 6 percent. Emission arrays for each link are used in the code to compute emissions that refer to the slope of the link.

1.4.1 Algorithms

The first problem is to extend the link into the intersection at the beginning of the link. The first step is to calculate the average speed over the first two segments on the link.

$$v_1 = \frac{\sum_{i=1}^{i=6} 7.5(i-1)N_{i1}}{\sum_{i=1}^{i=6} N_{i1}}$$

and

$$v_2 = \frac{\sum_{i=1}^{i=6} 7.5(i-1)N_{i2}}{\sum_{i=1}^{i=6} N_{i2}}$$

If v_2 is 25% greater than v_1, we use regression relationships to calculate the densities of the new first segment as:

$$N_{i0} = \sum_{k=1}^{k=4} \alpha_{ik} N_{k1} + \sum_{k=1}^{k=4} \beta_{ik} N_{k2}$$

with most of the α_{ik} and β_{ik} being zero. If v_2 is less than 25% greater than v_1, we use a linear extrapolation of the form:

$$N_{i0} = 2N_{i1} - N_{i2}$$

The second question that must be addressed is the fraction of the first speed bin density in the added segment that represents stationary vehicles that don't contribute to the flux. This fraction is estimated as:

$$fs(1) = N_{10} - {}^{N_{20}}\!/_{4}$$

With $fs(1)$ calculated, we refine N_{10} by subtracting $fs(1)$ from it to obtain the portion of the vehicles that participate in the flux of vehicles. Emissions for the stationary and flux participating vehicles are calculated separately and summed to find the total emissions.

The first step in estimating the velocity-acceleration distribution is to make a continuous fit to the densities (number of vehicles per unit speed and per unit space). A simple fit that is of the following form:

$$d_{ij}(\delta v) = f_{ij} + h_{ij}\delta v$$

where

$\delta v = v - j\Delta$ and *i* represents the spatial cell and *j* represents the speed bin (with Δ being the cell width, 7.5 meters or 7.5 meters per second for the speed bins).

The constant term f_{ij} is given by the following:

$$f_{ij} = N_{ij} / (4\Delta^2)$$

where the length of the box is 4Δ and the width in velocity space is Δ. The gradients slope terms (h_{ij}) are found by setting d_{ij} to zero at the top of the highest speed bin and solving for h_{ij}. Continuity relationships are used to determine the h_{ij}'s for the slower speed bins. However, this procedure may lead to negative densities over a portion of a speed bin.

This problem is solved in one of two ways:

1. If the negative value takes place in the slowest of the speed bins that have vehicles, the density relationship is assumed to hold down to a value of δv denoted δvi, where the density falls to zero and remains there. In other words, in the slowest, populated speed bin, the distribution extends from $\delta v=(\Delta/2)$ to $\delta v=\delta v_l$ rather than to $\delta v=-\Delta/2$.

2. The second situation takes place where the potential negative values occur in an intermediate speed bin. In this case, the continuity condition at the speed bin boundaries is relaxed and h_{ij} is set to zero.

Once the densities are known, the various moments of speed are calculated. Specifically, the following are calculated: the zeroth moment (average density), first moment (flux), second moment (needed for speed variance), and the third moment whose gradient is related to power.

The flux is divided into thirds and used to calculate the breakpoints between the slowest one-third ($j=n13$ and $\delta v=\delta v_l$), the middle one-third, and the fastest one-third ($j=n23$ and $\delta v=\delta v_h$). Once the breakpoints are determined, the third-moments of speed are calculated for each third of the flux. The probability of a hard acceleration is then estimated from the following:

$$P_a = \max(P_\sigma, P_{sp})$$

with

$$P_\sigma = p_0 + p_1 v_{avg}^2 (\sigma - \sigma_r)$$

where σ is the standard deviation of speed, v_{avg} is the average speed, and σ_r is the standard deviation of speed for an uncongested freeway.

The calibration constants p_0 and p_1 are determined separately for each third of the flux (slowest, intermediate, and fastest), but the standard deviation and the average speed refer to the entire vehicle flow in the segment. The foregoing treatment associated with the standard deviation of speed dominates when the average speed along the link does not change or is changing very slowly.

When there are significant speed changes along the link, the second component, P_{sp}, is given by

$$P_{sp} = p_{sp0} + p_{sp1} sp$$

with the speed gradient parameter, which in turn is defined by

$$sp = \frac{gradient - of - third - moment}{xeroth - moment \times \Delta^2}$$

dominates. In this case, the moments and the calibration constants are calculated for each third of the flux. The calibration constants were selected based on maximum gradients for vehicles leaving a signal on a fast arterial. In both the calibration and the emission estimation, *sp* is averaged over three adjacent cells.

The calibration constants were determined from a power balance (for each third of the vehicles: slowest, middle, and fastest), for high and low power. Regression relationships were developed of the following form:

$$Pow = Pow_0 + pow_1 sp$$

or, in the case of situations where $|sp|$ is small, the form was as follows:

$$Pow = pow_0 + pow_1 v_{avg}^2 (\sigma - \sigma_r)$$

The powers (*Pow*) refer to the continuous trajectories and are calculated on a per-trajectory basis. In our formulation, a single trajectory produces a flux of one-cell per second (or 16.7-mph). Consequently, the power for a given segment is given by the flux for that segment divided by 16.7-mph. The power of vehicles in the high-power mode is as follows:

$$\frac{flux}{16.7} Pow_h = p_h d \frac{(1 + e_o \alpha)}{\alpha} \ ,$$

where α is the exponent in cumulative distribution for power, e_o is the threshold for high-power driving, and d is the density of vehicles in the segment and the third of the vehicles under consideration.

With Pow_h provided by the regression relations, we can solve for P_h. We also can calculate the total power for each third of the vehicle flux as follows:

$$Pow_{tot} = \frac{1}{2} sp \ d \ \Delta^2$$

The average power of the intermediate driving is then estimated as follows:

$$\overline{Pow_m} = \frac{Pow_{tot} - Pow_h - Pow_l}{d(1 - P_h - P_l)}$$

The *pk* gives us the probability of a high-power event for the third of the flux under consideration. For a given speed, v_{ref} , the corresponding density is:

$$d_{ref} = {flux}/{v_{ref}}$$

and the probability of a high-power event is:

$$p_{ref} = \frac{Pow_k \bullet \alpha}{d_{ref} \bullet flux_{cor} \bullet [1. + (10. + 40 \bullet ef) \bullet \alpha]}$$

adjusted for both changes in α and the density associated with v_{ref} .

The population associated with a given aggressiveness, *ips*, index is then:

$$pop = [e^{-.3(ipa-1)} - e^{-.3ipa}] \bullet p_{ref} \bullet d_{ij}(\partial v) \bullet \Delta^2 / 8$$

with the associated power as:

$$pow = .3 \bullet (ipa - .5) / \alpha + e_o / \alpha - vsg$$

The term *vsg* is a correction for the power needed to drive the vehicle up the slope given by:

$$vsg = v_{ref} \bullet 21.95 \bullet \sin(slope / 57.3)$$

where slope is *grade* in percent of the link. The associated power index is:

$$ia = \frac{pow + 310.}{20} + 1$$

We next estimate the power at the previous second for a vehicle in the same power percentile. We first must determine whether or not a vehicle with the same power percentile will be in high-power driving or in the insignificant power driving range. In order for the vehicle to be in high-power driving, the probability of high-power driving for the previous position must be greater than:

$$p_{i-1} > p_i e^{-.3ipa}$$

in which case the previous power is estimated as:

$$Pow_{i-1} = .3ipa / \alpha - \log(p_{i-1} / p_i) + e_o / \alpha$$

If the test is not satisfied, the previous power is merely the average power in the intermediate power range.

There is an additional complication at the beginning of the link because the probability of high-power driving for the preceding segment is not defined. We assume that for the first second, the power, $pow_{sp=0}$, is the same as that for high-power driving with no net acceleration. If the speed is less than 4Δ and the vehicle is not in the highest flux group, we adjusted for the fraction of vehicles in the segment that have already spent one second in the cell. The fraction of the vehicles that spend their first second in the segment is obtained by integrating the density along the segment to v_{ref} and dividing by the integrated density over the entire segment. The result is:

$$\rho_{1\sec} = \frac{[(1.5\rho_1 - .5\rho_2)v_{ref} + \frac{(\rho_1 - \rho_2)v_{ref}^2}{8\Delta}]}{4\Delta\rho_1}$$

where ρ_1 is the average density of the first segment, and ρ_2 is the average density of the second segment. The average previous power is then:

$$pow_o = (1. - \rho_{1\sec})pow_p + \rho_{1\sec}pow_{sp=0}$$

In this expression, pow_p is given by:

$$pow_p = .3ipa/\alpha - \log(p_1/p_2) + e_o/\alpha$$

The fractional change in power is:

$$dpfac = \frac{(pow - pow_o)}{pow}$$

1.4.2 Scripts

TRANSIMS uses three scripts to construct the emission arrays. The first script produces the file *arrayp.out*. This script constructs input files for *cmemCore* from sample input for *cmemCore*; the files are *test-ctr* and *test-act*.

The new file *batch-ctr* provides the vehicle type and the soak time to *cmemCore*; it could also provide auxiliary power loads and changes in the vehicle parameters (if desired). The new file *batch-act* provides second-by-second speeds and grades for a four-second trajectory.

The third second of the trajectory is calculated for the desired power level and ending speed. The speeds for the preceding seconds are derived from a constant power level assumption. If this leads to unrealistically low speeds, we use a value of 0.1-mph. The script runs *cmemCore* with the following command:

```
% cmemCore batch> cntr.out
```

One of the output files from *cmemCore* is *batch-sbs*, which gives second-by-second emissions and fuel consumption. The script extracts the results from the third second of the trajectory and stores them in the file *batch.out*. There are three loops within the script:

- the outermost is on the vehicle type,
- the intermediate loop is on power (or more precisely, the product of speed and acceleration), and
- the innermost loop is on speed.

With *iv* the speed index and *ia* the power index, the speeds for each second of the trajectory are calculated as follows:

$$v(3) = 2. + 4 \times (iv - 1)$$

$$Pow = -300. + 20. \times (ia - 1)$$

$$v(2) = \sqrt{(v(3)^2 - 2 \times Pow)}$$

$$v(1) = \sqrt{(v(2)^2 - 2 \times Pow)}$$

$$v(4) = \sqrt{(v(3)^2 + 2 \times Pow)}$$

The file *batch.out*, which is renamed as *batchtotpc*, then contains the emissions for

- constant power with each line giving the speed,
- acceleration ($Acc = \dfrac{Pow}{v}$),
- hydrocarbon emissions,
- CO emissions,
- NO_x emissions,
- fuel consumption, and
- vehicle type.

The array parameter files are constructed by weighting the vehicle-type-specific emissions by the fraction of vehicles of that type in the fleet and summing over all vehicle types using the *CreateComposites* program.

In a similar fashion, the file *batchtotpj* is constructed, which gives the emissions by vehicle type when there is a step change in power between the second and the third seconds. The preceding equations apply, except that the speed for the first second is given by the following:

$$v(1) = v(2)$$

There is an additional complication when *ia* is less than 17: the calculations reflect a jump from positive power down to zero power. The starting power is:

$$pow = -300. + 20. \bullet (34 - ia - 1)$$

The velocities used are:

$$v(1) = \sqrt{v^2 - 2. \bullet pow}$$

and

$$v(2) = v = v(3) = v(4)$$

The resulting file *batch.out* is renamed *batchtotpj*. Another script takes the difference in emissions and fuel consumption between those in *batchtotpj* and those in *batchtotpc* and stores the results in the same format in a new file called *batchtotpd*. When *ia* is less than 17, the differences are calculated between those in *batchtotpj* and those of a file produced with constant speeds (zero power) rather than those with constant power. When weighted by vehicle type fraction and summed over all vehicle types, the result is the file are the array parameter files. The program *CreateComposites* is used to convert the batch* files to the array parameter files.

Similar scripts are generated for grades of 2%, 4%, and 6%. The scripts are identical except that the appropriate grade is used in the input to *cmemCore*. Note that the grade in *cmemCore* is in units of radians, so that:

$$grade_{cmemCore} = \frac{arcine(.01 \bullet grade)}{57.3}$$

with *grade* expressed in percent.

1.5 Heavy-Duty Vehicle (HDV) Tailpipe Emissions Estimator Submodule

The HDV Tailpipe Emissions Estimator submodule treats five weight classes of diesel-powered trucks and an additional diesel-powered bus. The actual emissions are based on relationships developed for diesel-powered buses in-lieu of data being developed for trucks under a National Cooperative Highway Research Program contract. HDV power demands are divided into four categories:

- deceleration,
- cruise or idle,
- modest power demands, and
- high-power demands.

The power bins are defined by fractions of full-throttle power for each speed. The HDV Tailpipe Emissions Estimator uses the same speed bins as does the LDV Tailpipe Emissions Estimator: 20 4-mph bins starting at 0-mph (represented by a speed of 2-mph).

For each HDV vehicle type, we have an array of maximum accelerations. Velocity-acceleration products less than minus one-tenth of the maximum product for a speed bin belong to the deceleration category. *V-A* products with absolute values less than one-tenth of the maximum *VA* product are defined as cruise, whereas *VA* products greater than 25% of the maximum *VA* product are defined as high-power demands. Fig. 5 depicts the HDV emission flow.

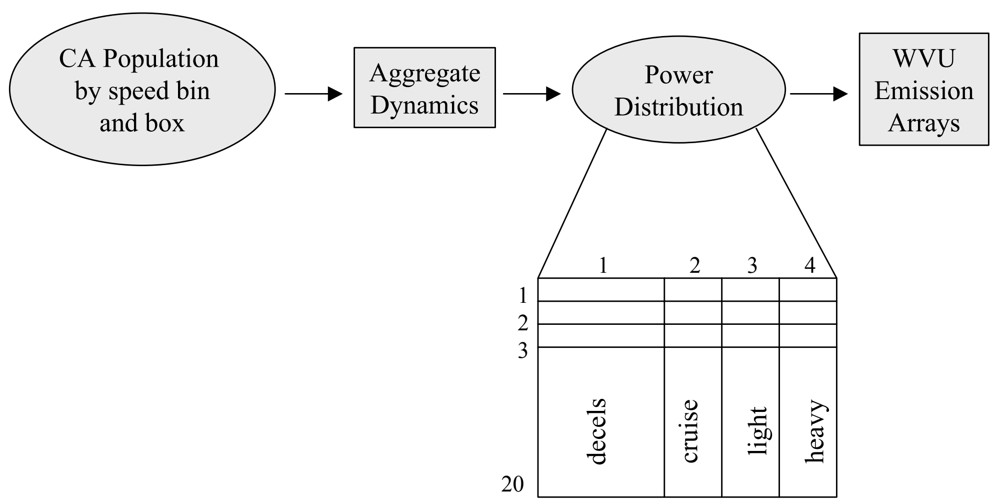

Fig. 5. An overview of the HDV Tailpipe Emissions Estimator submodule.

There are a number of important differences between HDVs and LDVs. For example, the power distribution is much different for HDVs than it is for LDVs. Fig. 6 reports the cumulative power distribution for a heavy tractor-trailer.

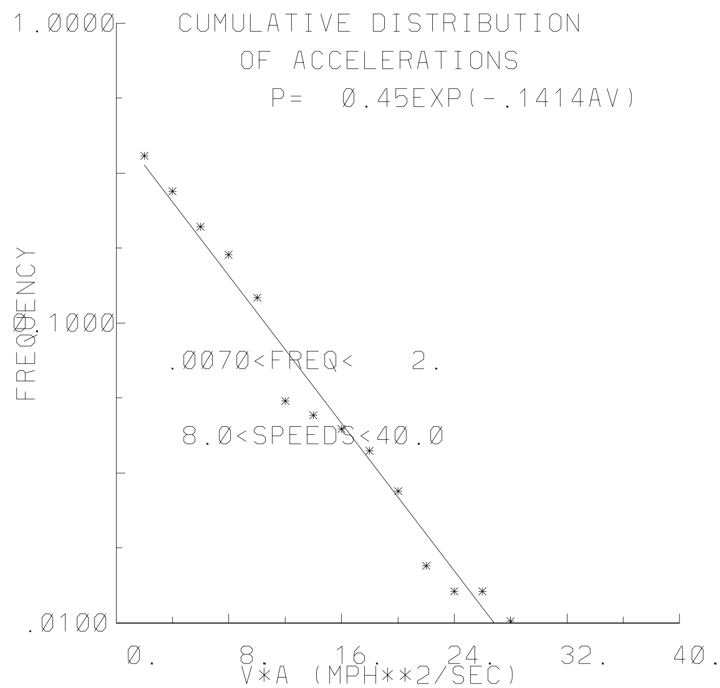

Fig. 6. A cumulative distribution of accelerations.

There is still a very good fit for an exponential decay of frequency with *VA* product, but the decay is much faster. The second important difference is that the emissions are more sensitive to speed but less sensitive to the magnitude of power demands. Fig. 7 reports bus NO_x emissions as a function of speed for different fractional power demands.

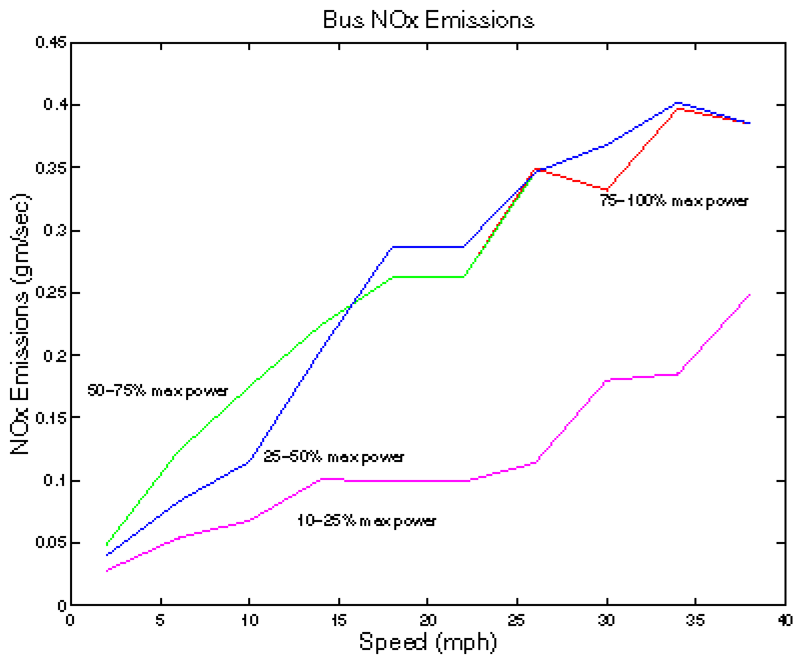

Fig. 7. Bus NO_x emissions.

Fig. 7 supports the utility of a representation using two positive power levels. The aggregate dynamics module functions much like the one used for LDVs. For this module, its first step is to produce a continuous speed distribution for each box along the link. This approach mirrors that used for the LDVs.

There are two difficulties that arise when we attempt to follow the approach used for the LDVs to define the population of the four power bins for each speed bin.

- First, the power bins for HDVs differ from one speed bin to the next because they are defined in terms of the maximum acceleration for the speed bin.
- Second, we don't have collections of context-specific trajectories to use to estimate important parameters.

Despite these differences, we expect that there remain strong similarities between the behavior of LDVs and HDVs. Specifically, congestion waves will drive high-power driving, as will the need to regain speed-after-speed reductions at traffic control points. Consequently, the fraction of time that higher power driving takes place is influenced by the same factors for HDVs as it is for LDVs. Specifically, the standard deviation of speeds and the gradient of the cube of the speed should dictate the fraction of the driving that consists of high-power driving.

Unfortunately, we cannot use the same coefficients because the basic power distributions are different and the power bins are defined differently. However, we can find adjustments to the LDV coefficients that reflect the differences in power distribution and power bins.

The first step is to define an HDV power level that corresponds to the (50-mph squared per second squared) power level that separates high-power driving from other driving for LDVs. Looking at all driving for LDVs, we find that power demands of 50-mph squared per second squared take place approximately 6.8% of the time. With the relationship developed from Fig. 6, we find that the corresponding power level for HDVs is 13.5-mph squared per second squared.

Consequently, for similar driving conditions we would expect the probability that the LDVs exceed a power demand of 50-mph squared per second squared is similar to the probability that HDVs exceed a power demand of 13.5-mph squared per second squared. We also would expect that, because *sp* is proportional to power, the *sp*s deduced for HDVs would be (13.5/50.) times the *sp*s deduced for LDVs. Equating the probabilities enables us to estimate the coefficients for HDVs from those of LDVs.

The result is a method to estimate the probability that HDV power demands exceed 13.5-mph squared per second squared. Once that probability is estimated, we can use the exponential relationship of Fig. 6 to estimate the population in the 10-25% maximum power bin and in the 25-100% maximum power bin.

1.5.1 Algorithms

The probability of high-power driving with low *sp* is estimated with the following:

$$p_{\sigma h} = p_{0h} + p_{1h} v_{avg}^2 (\sigma - \sigma_r)$$

whereas the constant p_{0h} is estimated as follows:

$$p_{0h} = p_0 \left(\frac{\alpha}{1+50\alpha} \right) \left(\frac{1+13.5\alpha_h}{\alpha_h} \right)$$

and the constant p_{1h} is estimated as follows:

$$p_{1h} = p_1 \left(\frac{\alpha}{1+50\alpha} \right) \left(\frac{1+13.5\alpha_h}{\alpha_h} \right)$$

with α appropriate for general driving of LDVs and α_h drawn from Fig. 6 (0.141).

In the case where *sp* is important, the probability of high-power driving is estimated with the following equation:

$$P_{sph} = P_{sp0h} + p_{sp1h}(sp)$$

where p_{sp0h} is estimated by the following:

$$p_{sp0h} = p_{sp0} \left(\frac{\alpha}{1+50\alpha} \right) \left(\frac{1+13.5\alpha_h}{\alpha_h} \right)$$

whereas p_{sp1h} is estimated by the following:

$$p_{sp1h} = p_{sp1}\left(\frac{\alpha}{1+50\alpha}\right)\left(\frac{1+13.5\alpha_h}{\alpha_h}\right)\left(\frac{50}{13.5}\right)$$

with the factor $\left(\frac{50}{13.5}\right)$ used to adjust for the differences between *sps* for LDVs and *sps* for HDVs.

To calculate the fraction of vehicles in a specific power bin, we first calculate the appropriate power. For the 10 to 25% power bin, the lower limit is as follows:

$$Pow10 = 0.1va_{\max}$$

whereas the upper limit is as follows:

$$Pow25 = 0.25va_{\max}$$

and the fraction of the high-power driving that falls between the limits is as follows:

$$\frac{e^{-\alpha_h pow10} - e^{-\alpha_h pow25}}{e^{-\alpha_h 13.5}}$$

with corresponding expressions for the fraction in the 25 to 100% power bin.

1.6 Usage

Much like EPA's MODELS-3, the Tailpipe Emissions Estimator is designed to produce emissions appropriate for simulations of air quality over a metropolitan area. It does not produce intersection emissions in this version.

It is designed to produce fleet average emissions rather than emissions from individual vehicles. It uses continuous approximations for densities, and thus requires that many vehicles be considered over at least a one-hour period in order for appropriate statistics to be developed.

Use the *EmissionsEstimator* program for the Light Duty Vehicle Tailpipe emissions model and the *EmissionsEstimatorHDV* program for the Heavy Duty Vehicle Tailpipe model.

Usage:

```
EmissionsEstimator <configFilename> [<extension>]
```

where

`<configFilename>` is the name of the configuration file used to run the Traffic Microsimulator to collect output

`[<extension>]` is an optional argument used when the postprocessed velocity output has been distributed to more than one velocity file by the *distribVELfile* utility program

Usage:

```
EmissionsEstimatorHDV <configFilename>
```

where

`<configFilename>` is the name of the configuration file used to run the Traffic Microsimulator to collect output

1.7 Version Notes

There are a number of assumptions in the current version of the code:

- The number of 30-meter segments on a link populated with vehicles must be at least two—both of which must have vehicles.
- There can be empty cells before and after a set of populated ones, but there must not be any unpopulated segments interspersed with populated ones.
- The speed histograms should be constructed from one-second sampling, summed over one hour or more. The direction of travel on the links is important.

1.8 Major Input/Output

Fig. 8 summarizes the data flow of the Tailpipe Emissions Estimator. The Tailpipe Emissions Estimator requires information on the fleet composition, which is developed from the Population Synthesizer, vehicle loads, and traffic patterns. The Population Synthesizer provides vehicle fleet characteristics, including the fraction of the fleet that is malfunctioning. The Traffic Microsimulator produces the traffic patterns.

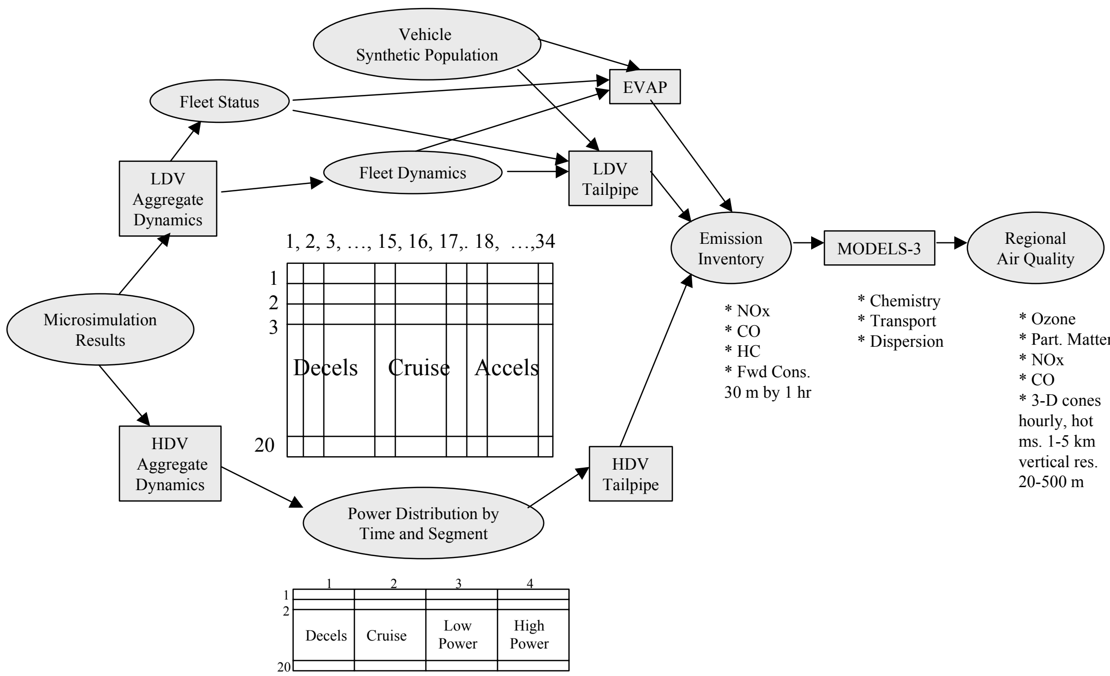

Fig. 8. The Tailpipe Emissions Estimator data flow.

The Traffic Microsimulator produces the bulk of the information required by the Tailpipe Emissions Estimator. The following items are included in this category:

- spatial summaries of vehicle velocities over 30-meter sections of roadway,
- histograms of the number of vehicles entering a link grouped by velocity-acceleration product and by time since the vehicles were last parked (soak time).

Tailpipe emissions output data is aggregated on 30-meter segments for each simulated one-hour period. Fuel economy and CO_2 emissions are also estimated. The emission inventory is designed to be used with the MODELS-3 code, which was developed by the EPA to produce three-dimensional, hourly, gridded emissions values over the metropolitan area. Appendix A lists the TRANSIMS configuration file keys specific to the Tailpipe Emissions Estimator.

Appendix B lists the TRANSIMS configuration file keys that must be set to a specific value for the Tailpipe Emissions Estimator to calculate emissions correctly.

1.8.1 Tailpipe Emissions Estimator Files

The Tailpipe Emissions Estimator calculates emissions in 30-meter segments along a link for selected time periods (normally 1 hour). It gives estimates of tailpipe emissions of nitrogen oxides (NO_x), carbon monoxide (CO), and hydrocarbons from LDVs. and HDVs It also gives fuel consumption that can be used to calculate emissions of carbon dioxide (CO_2). The file names seen below are defined in the code as default names but may be changed by using configuration file keys. See Appendix A for a list of those configuration file keys.

1.8.1.1 LDV Files

This section describes file formats for each of the 17 input files (12 fairly static and five changing files) for the LDV Tailpipe Emissions Estimator and the three output files that it may produce.

1.8.1.1.1 *ARRAYP.INP* File

The *ARRAYP.INP* file is used in conjunction with *arrayp*.out* and *arraypd*.out* files that contain parameters describing the number of records and increments used in these files. Several fields are unused by the LDV Tailpipe Emissions Estimator. Appendix C-1 describes the file's fields.

1.8.1.1.2 *arrayp*.out* and *arraypd*.out* Files

The *arrayp*.out* file gives composite vehicle emissions in 4-mph speed bins and 20-mph squared per second power bins.

- The *arrayp.out* file gives composite vehicle emissions for street grades of less than 1% and downhill grades.
- The *arrayp2p.out* file gives composite vehicle emissions for street grades between 1% and 3%.
- The *arrayp4p.out* file gives composite vehicle emissions for street grades between 3% and 5%.
- The *arrayp6p.out* file gives composite vehicle emissions for street grades above 5%.

The *arraypd*.out* file gives the difference in emissions between constant power trajectories and those with the same speed and power but with a step change in power. The *arraypd.out* file is for street grades of less than 1% and downhill grades. Similar to *arrayp.out* files, the 2%, 4%, and 6% files are associated with those street grades. All of these files are input files for the LDV Tailpipe Emissions Estimator. The first two lines of each file contain headers information that is ignored. Only the data fields are described in Appendix C-2.

1.8.1.1.3 ****SoakRatios* File**

The three **SoakRatios* files (*shortSoakRatios*, *mediumSoakRatios*, and *longSoakRatios*) are input files in the LDV Tailpipe Emissions Estimator and contain the composition ratios of cold emissions to hot engine emissions for each of eight power levels and for each of the emissions components. These three files contain ratios based on the assumption that the soak levels were set to specific values with the `CA_*_SOAK_TIME` configuration file keys when the simulation was run. The default (and assumed values) for these configuration file keys can be found in Appendix B. Appendix D describes the fields in the **SoakRatios* files.

1.8.1.1.4 ***velocity.ldv.out* File**

The *velocity.ldv.out* file contains the link velocity summary data produced by the Traffic Microsimulator and reformatted for input into the LDV Tailpipe Emissions Estimator. The transformation is performed by using the *ConvertVELfile* program.

The first six items described in Appendix E (`NV` through `DISTANCE2`) appear in a single record, followed by `NV` records containing the six `COUNT` fields in order of each record. This sequence is repeated for each `LINK`, `NODE`, and `TIME` step in the original file.

1.8.1.1.5 ***energy.*.out* File**

The four *energy.*.out* files (*energy.no.out*, *energy.short.out*, *energy.medium.out*, and *energy.long.out*) contain the distribution of LDVs entering the link, stratified by the time-integrated, velocity-acceleration product for each of the file's soak times (the time the engine was idle before the start of the current trip). These files are created by the *ConvertENRfile* program using the four microsimulation energy summary files. See Appendix F for a field descriptions.

- The *energy.no.out* file contains power distributions for vehicles with a soak time less than `CA_SHORT_SOAK_TIME`.
- The *energy.short.out* file contains power distributions for vehicles with a soak time between `CA_SHORT_SOAK_TIME` and `CA_MEDIUM_SOAK_TIME`.
- The *energy.medium.out* file contains power distributions for vehicles with a soak time between `CA_MEDIUM_SOAK_TIME` and `CA_LONG_SOAK_TIME`.
- The *energy.long.out* file contains power distributions for vehicles with a soak time greater than or equal to `CA_LONG_SOAK_TIME`.

The default (and assumed values) for these soak time configuration file keys are described in Appendix B.

The configuration file key `OUT_SUMMARY_ENERGY_BINS_DEFAULT` or `OUT_SUMMARY_ENERGY_BINS_n` determines the number of energy bins in each energy summary file. These files must contain data for eight bins, so the configuration file keys must be set accordingly (the negligible soak file set by `OUT_SUMMARY_ENERGY_SOAK_n NEGLIGIBLE` may contain only one energy bin, if desired).

The configuration file key `OUT_SUMMARY_ENERGY_MAX_DEFAULT` or `OUT_SUMMARY_ENERGY_MAX_n` sets the bounds to the energies for each energy bin. The current default (and assumed value) for this configuration file key is `105` (except for the no soak, which may be 0). The above configuration file keys are described in Appendix B.

Note that negative accelerations are ignored in the calculation of the time-integrated, velocity-acceleration products. This distribution is used to determine which cold/warm emission ratios should be used.

1.8.1.1.6 *emissions.ldv.out* File

The *emissions.ldv.out* file is the final output file produced by the LDV Tailpipe Emissions Estimator. This file is written using the variable-size box format and is ready to be visualized with the Output Visualizer. Each record contains the five fields required by this format, plus six data values (as described in Appendix G).

1.8.1.1.7 *debug.ldv.out* and *calcsum.ldv* File

Output files produced by the LDV Tailpipe Emissions Estimator (*debug.ldv.out* and *calcsum.ldv*) are debugging files that provide intermediate output for the emission calculations. These two files will be produced only if the configuration file key `EMISSIONS_WRITE_DEBUG_OUTPUT` is set to 1.

Appendix H provides fields in the output for *debug.ldv.out*. Appendix I provides fields in the *calcsum.ldv* debugging output file.

1.8.1.2 HDV Files

This section describes file formats for each of the three input files for the HDV Tailpipe Emissions Estimator and the three possible output files it produces.

1.8.1.2.1 *ARRAY_HDV.INP* File

The *ARRAY_HDV.INP* file is used in conjunction with the *catruck.acc* file and contains parameters describing the number of records and increments used in these files. Several fields are unused by the HDV Tailpipe Emissions Estimator. Appendix C-1 provides these fields.

1.8.1.2.2 *catruck.acc* File

The *catruck.acc* file gives the typical emissions and fuel consumption at constant power for different levels of power and speed for a single grade. This file contains 20 rows of single values that represent levels for accelerations and decelerations for various levels of aggressiveness and speeds. Two lines of text are then ignored followed by composite emissions for 20 speed bins and four power bins for nitrogen oxides (NO_x), carbon monoxide (CO), and hydrocarbons along with particulate matter and fuel consumption.

1.8.1.2.3 *velocity.hdv.out* File

The *velocity.hdv.out* file contains the link velocity summary data produced by the Traffic Microsimulator and reformatted for input into the HDV Tailpipe Emissions Estimator. The transformation is performed by using the *ConvertVELfile* program.

The first six items described in Appendix E (`NV` through `DISTANCE2`) appear in a single record, followed by `NV` records containing the six `COUNT` fields in order of each record. This sequence is repeated for each `LINK`, `NODE`, and `TIME` step in the original file.

1.8.1.2.4 *emissions.hdv.out* File

The *emissions.hdv.out* file is the final output file produced by the HDV Tailpipe Emissions Estimator. This file is written using the variable-size box format and is ready to be visualized with the Output Visualizer. Each record contains the five fields required by this format, plus six data values (as described in Appendix G).

1.8.1.2.5 *debug.hdv.out* and *calcsum.hdv* Files

Output files produced by the HDV Tailpipe Emissions Estimator (*debug.hdv.out* and *calcsum.hdv*) are debugging files that provide intermediate output for the emission calculations. These two files will be produced only if the configuration file key `EMISSIONS_WRITE_DEBUG_OUTPUT` is set to 1.

Appendix H provides fields in the output for *debug.hdv.out*. Appendix I provides fields in the *calcsum.hdv* debugging output file.

1.9 Utility Programs

1.9.1 ConvertVELfile Utility

The *ConvertVELfile* program transforms the microsimulation link-velocity summary output into the format required by the Tailpipe Emissions Estimator. The link is partitioned into boxes of a constant size, except that the last box on the link may be shorter than the others.

The *ConvertVELfile* program proportionally inflates the values for the last box to what might be expected if the box were full sized.

Note that *ConvertVELfile* includes some assumptions that are more restrictive than the generality in the output available from the Traffic Microsimulator. For example, the program assumes that the boxes that partition the link are 30-meters long; a value other than 30 for the microsimulation configuration file key `OUT_SUMMARY_BOX_LENGTH_DEFAULT` or `OUT_SUMMARY_BOX_LENGTH_n` used when collecting velocity data will result in velocity summary data that cannot be correctly processed by *ConvertVELfile*.

The ConvertVELfile program assumes that the `OUT_SUMMARY_VELOCITY_MAX_DEFAULT` or `OUT_SUMMARY_VELOCITY_MAX_n` configuration file key is set to 37.5 (the maximum microsimulation velocity). The *ConvertVELfile* program assumes that exactly six velocity histogram bins are defined. The simulation must be run with the configuration file key `OUT_SUMMARY_VELOCITY_BINS` set to 5 in order for this to be accomplished. An overflow bin will be created automatically. There are a few other configuration file keys that need to be set to ensure proper processing (such as file names, vehicle types, and sample and output timesteps). See Appendix B for these configuration file keys.

1.9.2 ConvertENRfile Utility

The *ConvertENRfile* program transforms the microsimulation link-energy summary output into the distributions required by the Tailpipe Emissions Estimator. This utility assumes all four types of energy files were collected and will attempt to convert all four. The `OUT_SUMMARY_ENERGY_SOAK_n` configuration file key is used to specify, for each file, which type of soak collection. The possible values for this configuration file key are `NEGLIGIBLE`, `SHORT`, `MEDIUM`, and `LONG`. To set the soak levels, the configuration file keys `CA_SHORT_SOAK_TIME`, `CA_MEDIUM_SOAK_TIME`, and `CA_LONG_SOAK_TIME` are set before the simulation is run. See Appendix B for the values these should be set to. The *ConvertENRfile* program assumes that the `OUT_SUMMARY_ENERGY_MAX_DEFAULT` or `OUT_SUMMARY_ENERGY_MAX_n` configuration file key has been set to `105` (or `0` for the no soak file).

The *ConvertENRfile* program assumes that exactly eight velocity histogram bins are defined. The simulation needs to be run with the configuration file key `OUT_SUMMARY_ENERGY_BINS_n` set to `7` in order for this to be accomplished. An overflow bin will be created automatically.

1.9.3 CreateComposites Utility

The *CreateComposites* program takes in files containing the array of composite emissions and combines the vehicle types and outputs composite emissions by speed and power for a single vehicle type. Representative emissions are for 20 speed levels, 34 power levels, and 23 vehicle types for light-duty vehicles (speed bins of 4-mph and power at intervals of 20). *CreateComposites'* inputs are *batchtopc** (or *batchtopd**) and the *vehdist* file. Its outputs are the *arrayp*.out* and *arraypd*.out* files. The *vehdist* file contains the distribution of the 23 LDV subtypes. Appendix J lists fields in the vehicle type distribution file.

1.9.4 ConvertTRVfile Utility

☛ This feature is not currently in use.

The *ConvertTRVfile* program can be used to convert a traveler events output file into a file containing the vehicle distributions (if that data is not available from the *vehdist* file).

1.9.5 DistribVELfile Utility

The distribVELfile program takes a postprocessed velocity file (*velocity.ldv.out*) and distributes it to several separate files by timestep. The resulting file names are the original velocity file name with an extension starting at .AA through the number of file creations requested.

This utility can be used to run multiple copies of the *EmissionsEstimator* code in order to speed up the running of the Light Duty Vehicle Tailpipe emissions step.

<u>Usage</u>:

```
distribVELfile <inputfilename> <#outputFiles> <startTime> <endTime>
```

where

`<inputfilename>` is the name of the file created from *ConvertVELfile*

`<#outputFiles>` is the number of files to divide the velocity output into

`<startTime> <endTime>` is the time range, in seconds, to include in the distributed velocity files.

<u>Example</u>:

```
distribVELfile velocity.auto.out 24 0 86400
```

After distribution of the velocity files, the script *RunLDVEmissions* may be run to start up all of the *EmissionsEstimator* processes. Once completed, the output files may be combined with the *combineEmissions* utility program.

1.9.6 RunLDVEmissions script Utility

This script is used to run the LDV EmissionsEstimator on more than one node.

<u>Usage</u>:

```
RunLDVEmissions <config file> <#files>
```

where

`<config file>` is the configuration file used as an argument to *distribVELfile*

`<#files>` is the number of files specified in the call to *distribVELfile*

1.9.7 combineEmissions Utility

The *combineEmissions* program takes several emissions output files and combines them together into one file. This can be used when multiple copies of the *EmissionsEstimator* code are run on separate (distributed) velocity files.

Usage:

```
combineEmissions <baseFilename> <#files>
```

where

`<baseFilename>` is the name of the emissions output file without the extension

`<#files>` is the number of files specified in the call to RunLDVEmissions

Example:

```
combineEmissions emissions.auto.out 24
```

1.10 Files

Appendix L provides library files for the Tailpipe Emissions Estimator.

Appendix M provides examples that used the calibration 2 network, which is the intersection calibration network.

1.11 Examples

Fig. 9 through Fig. 14 show examples of emission visualizations calculated for the intersection calibration network.

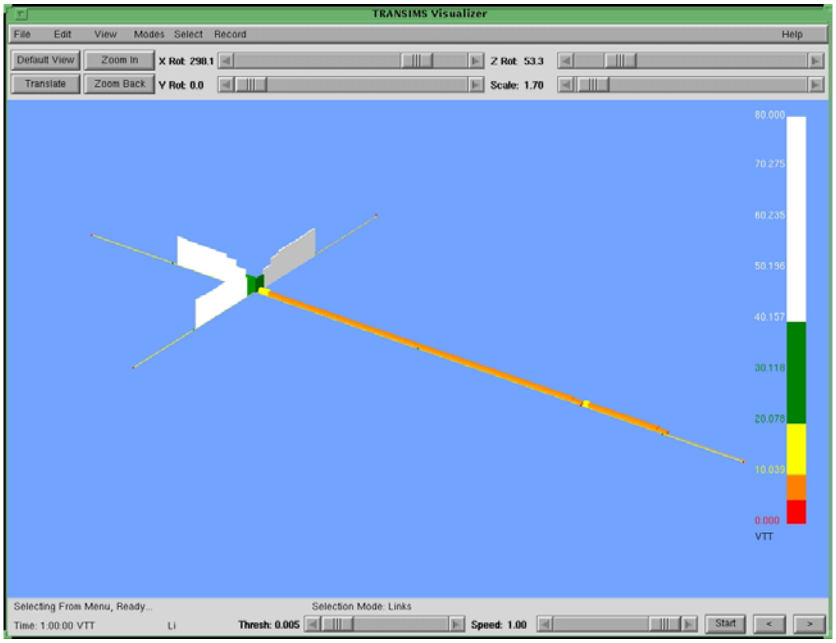

Fig. 9. This graphic depicts velocities.

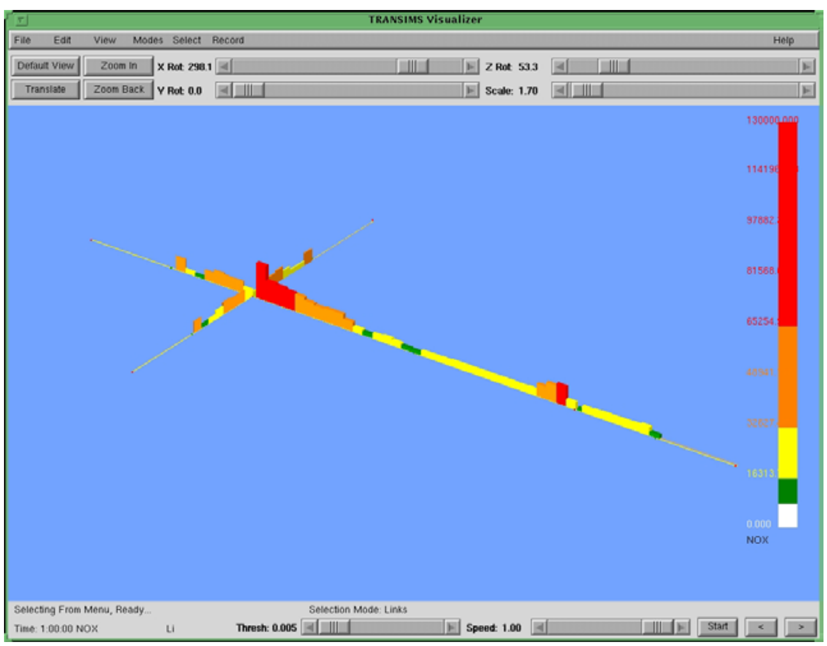

Fig. 10. This graphic depicts NO_x (nitrogen oxides) emissions.

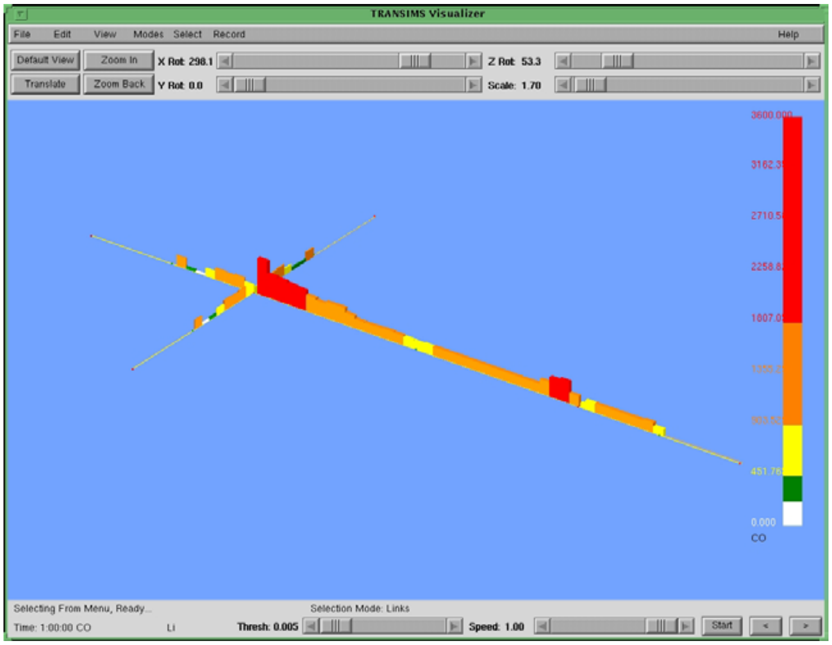

Fig. 11. This graphic depicts CO (carbon monoxide) emissions.

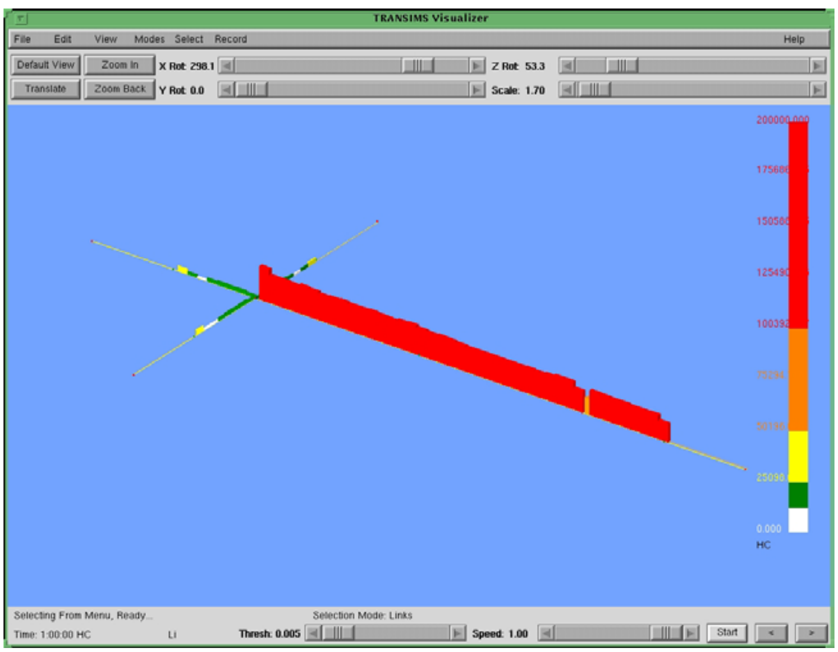

Fig. 12. This graphic depicts HC (hydrocarbon) emissions.

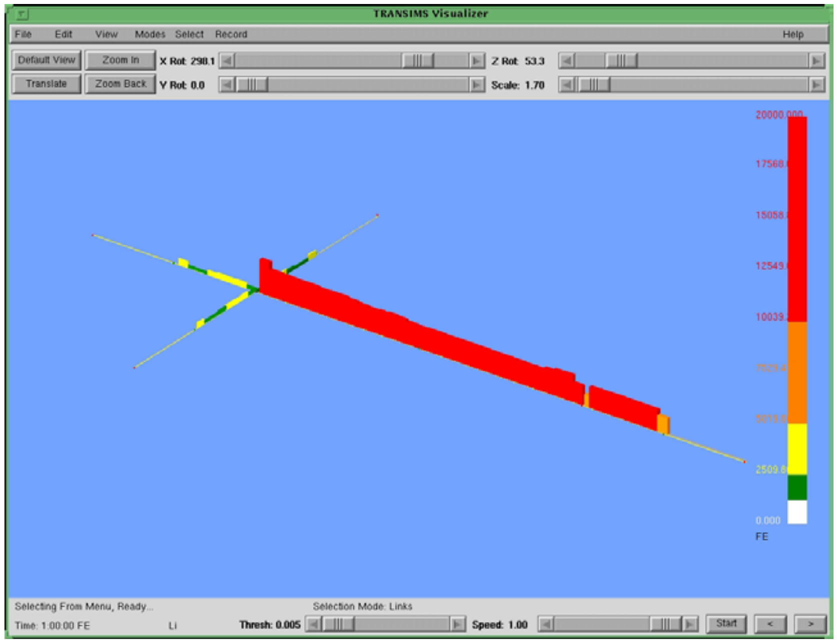

Fig. 13. This graphic depicts FE (fuel consumption).

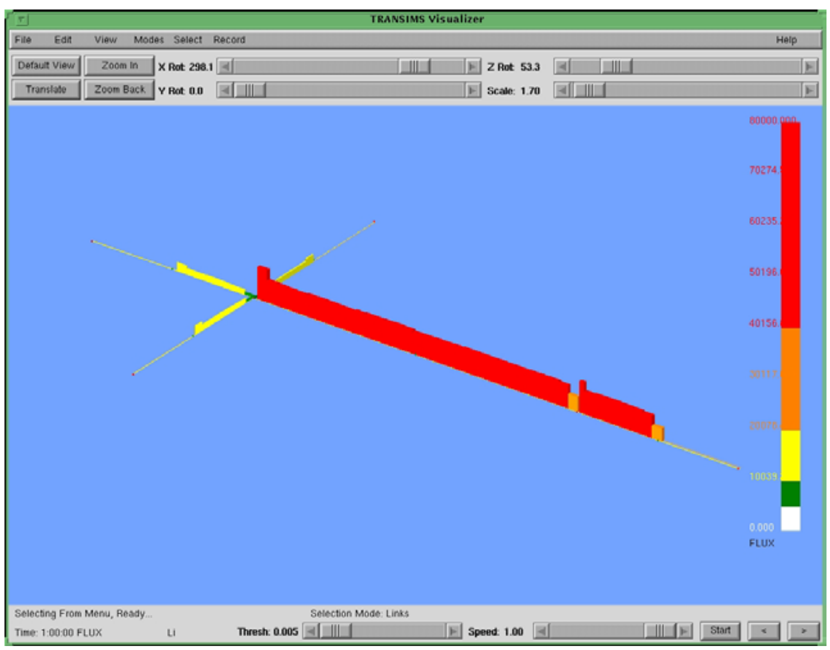

Fig. 14. This graphic depicts FLUX (vehicle flux).

2. Evaporative Emissions

2.1 Overview

Evaporative emissions consist of emissions that are caused by the evaporation of the fuel in vehicles. There are four mechanisms that contribute to the evaporative emissions:

- resting losses,
- diurnal emissions,
- hot soak, and
- running losses.

2.1.1 Resting Losses

Resting losses emissions result from fuel migrating through plastic hoses, gas tanks, and fittings. These emissions are dependent on the volatility of the fuel and the ambient temperature. They take place during all states of vehicle operation.

2.1.2 Diurnal Emissions

Diurnal emissions are caused by temperature changes that take place during the day. As the ambient temperature increases, more fuel in the gas tank vaporizes, thus increasing the pressure. The increased pressure is vented past a carbon filter that removes most of the fuel vapors, but it does not remove them all. As a vehicle is operated, air is drawn past the filter purging the filter. Diurnal emissions occur only during the portion of the day when the ambient temperature is increasing.

2.1.3 Hot Soak

Hot soak emissions occur for an hour after a vehicle has been running but has been turned off. These emissions are caused by the fuel resident in fuel lines, gas tank, carburetors, fuel injectors, etc., vaporizing due to the hot engine and leaking through fittings, migrating through plastic hoses, or vaporizing in the carburetor bowl.

2.1.4 Running Losses

Running loss emissions occur when a vehicle is operating. They are thought to be caused by two factors:

- increased resting losses because the fuel system is at a higher temperature than ambient, and
- pressurization of the fuel system and the resultant leaks at fittings and connections.

The proposed EPA model for running loss emissions for MOBILE6 assumes that two mechanisms account for the emissions. The first is a heating of the fuel system when the vehicle is operating, resulting in higher emission for longer trips. The other is an incomplete purge of the carbon filter at low speeds resulting in higher emissions at low speeds. The TRANSIMS running loss emissions model uses this EPA model.

TRANSIMS describes the vehicle movements that are translated into evaporative emissions with the EPA methodology. The EPA methodologies used in TRANSIMS are the preliminary methodologies for MOBILE6 reported by EPA. Any revisions of the methodologies done by the EPA for the final version of MOBILE6 are not incorporated in TRANSIMS.

2.2 The EPA Model System

The EPA divides LDVs into 16 classes. The first division is by vehicle age. Vehicles are divided into

- pre-1980 vehicles,
- vehicles from 1980 to 1986, and
- vehicles newer than 1986.

The second division is whether a vehicle fails or passes a pressure and a purge test. The divisions are as follows:

- vehicles that fail the pressure test,
- vehicles that fail only the purge test, and
- vehicles that pass both tests.

The last division is whether a vehicle is fuel injected or has a carburetor. The Model assumes that all pre-1980 vehicles have a carburetor. There is one additional class that is used by EPA—gross liquid leakers, or vehicles that have liquid leaks of fuel.

The EPA model provides formulas to calculate the number of leakers, number of vehicles that pass both the pressure and purge tests, and the number of vehicles that fail the pressure test based on the age of the vehicle[10]. The evaporative emissions are then calculated using formulas with different coefficients for the various vehicle classes.

2.2.1 Resting Loss

The resting loss calculation uses a single formula based on fuel RVP (Reed Vapor Pressure volatility measurement) and ambient temperature[11]. The resting losses are assumed to take place at all times, except when the vehicle is operating and when the

[10] Landman, Larry C. "Estimating Weighting Factors for Evaporative Emissions in MOBILE6", EPA420-P-99-023, U.S. Environmental Protection Agency, Ann Arbor, MI, (June 1999).
[11] Landman, Larry C. "Evaluating Resting Loss And Diurnal Evaporative Emissions Using RTD Tests, EPA420-P-99-026, U.S. Environmental Protection Agency, Ann Arbor, MI (July 1999).

vehicle is in hot soak. A single value of 9.16 grams per hour is used for the emissions of leakers for resting losses.

2.2.2 Diurnal Emission

The diurnal emission calculation first calculates the total diurnal emissions for a 24-hour period[12] using different coefficients for the various vehicle classifications. Leakers are assumed to emit 104.36 grams per day of evaporated fuel due to diurnal emissions. The next step is to calculate the partial diurnal emissions for vehicles that are started and stopped during the day[13]. The emissions caused by a partial diurnal are assumed to take place only for the period of time two hours after the vehicle is turned off and when the ambient temperature is higher than the ambient temperature two hours after the vehicle was turned off. Each vehicle classification has its own formula to calculate the partial diurnal emission.

2.2.3 Hot Soak Emissions

Hot soak emissions are assumed to take place one hour after the vehicle has been turned off. Calculated first are the total emissions for the one-hour[14]. A different classification scheme for the vehicles is used for hot soak. The categories used are as follows:

- vehicle with carburetors,
- PBI fuel-injected vehicles,
- cars,
- light-duty trucks,
- pre 1980 vehicles,
- 1980 to 1986 vehicles,
- vehicles newer than 1986,
- and leakers.

The last step consists of

1) calculating the distribution of the total one-hour hot soak over the hour, and
2) determining the emissions for vehicles that are restarted in less than an hour. This is done using an emissions rate versus time provided by the EPA[15].

[12] Landman, Larry C. "Evaluating Resting Loss And Diurnal Evaporative Emissions Using RTD Tests, EPA420-P-99-026, U.S. Environmental Protection Agency, Ann Arbor, MI (July 1999).
[13] Landman, Larry C. "Modeling Hourly Diurnal Emissions and Interrupted Diurnal Emissions Based on Real-Time Diurnal Data' EPA420-P-99-027, U. S. Environmental Protection Agency, Ann Arbor, MI (July 1999).
[14] ARCADIS Geraghty & Miller, Inc. " Update of Hot Soak Emissions" EPA420-P-99-005, U.S. Environmental Protection Agency, Ann Arbor, MI (Feb 1999).
[15] Glover, Edward L. "Hot Soak Emissions as a Function of Soak Time" EPA420-P-98-018, U.S. Environmental Protection Agency, Ann Arbor, MI (June 1998).

2.2.4 Running Loss

The running loss emissions calculation used by EPA is the MOBILE5 model, which assumes that running loss emissions are dependent on the trip length and average speed of the vehicle.

The total emissions will be calculated using this MOBILE5 methodology[16]. One difference from the MOBILE5 methodology (the standard EPA code for calculation of emission inventories for cities for regulatory purposes) will be the inclusion of leakers in the running-loss calculations, assuming an emission rate of 17.65 grams per mile. Values are given for emissions in grams per hour for different trip lengths and trip speeds.

2.3 The Evaporative Emissions Estimator Module

The TRANSIMS Evaporative Emissions Estimator takes vehicle activity information (start time and parking location, end time and parking location, and whether the vehicle is operating at the beginning of the simulation) to determine non-operating evaporative emissions and their locations. It closely follows the EPA MOBILE6 methodology.

The evaporative emissions for the period of time the vehicle is operating are found by calculating an evaporative emission factor, which then is multiplied times the total number of cells that are occupied on a link over a 1-hour time segment. This factor will depend on the fleet mixture, trip lengths, and average speeds for the link during the 1-hour interval. This module contains seven C++ program classes that are use primarily for storing constants or fleet information. The following sections describe these classes.

2.3.1 Algorithm

2.3.1.1 Coefficient Class

The `Coefficient` class contains formula coefficients for diurnal emissions calculations. It has **`get`** functions.

2.3.1.2 Record Class

The `Record` class contains the activity record for a vehicle. The constructor opens the activity data file and reads in the first record. This class has **`get`** and **`set`** functions.

2.3.1.3 Temperature Class

The `Temperature` class contains the hourly temperatures. Hour temperatures are loaded from a city-specific data file, and then minimum and maximum temperatures for the day are determined. This class has **`get`** functions that will wrap for one hour at the beginning and end of the day.

---

[16] Landman, Larry C, "Estimating Running Loss Evaporative Emissions in MOBILE6", EPA420-P-98-024 U.S. Environmental Protection Agency, Ann Arbor, MI (June 1999).

2.3.1.4 UserDefined Class

The `UserDefined` class contains the following fleet information:

- model year distribution of vehicles,
- percentage of vehicles that are cars versus pickups,
- percentage of vehicles having carburetors (by year),
- percentage of vehicles with TBI fuel injection systems (by year),
- RVP of fuel, and
- present year.

This is information that will be supplied by the MPO for the area. The data are loaded in from a city-specific data file.

2.3.1.5 Vapor-Pressure Class

The `Vapor-Pressure` class contains vapor pressure calculations and is dependent on the RVP of the fuel. The constructor calculates coefficients for the RVP. This class has **`set`** functions that calculate max and min vapor pressures depending on temperatures. It also has **`get`** functions.

2.3.1.6 Vehicle Class

The `Vehicle` class contains information about the vehicle and its activities that are being calculated. The information is as follows:

- vehicle ID,
- vehicle evaporation type,
- number of records in the vehicle activity file,
- model year of the vehicle,
- time of day when the vehicle was turned on or off,
- whether the vehicle is on or off,
- location of the vehicle,
- if the vehicle is a car, and
- if the vehicle has TBI fuel injection.

The `Vehicle` class has **`get`** and **`set`** functions.

The function **`main`** contains the loop that reads all vehicle activity records and calculates the emissions and their locations (for non-operating vehicles) in 15-minute segments for the time of the simulation.

- The first function call is to **`readActivityFile`**, which reads the activity file and enters the information in a `Record` class until an EOF (end-of-file) is encountered.
- The second function call is to **`getVehicleActivity`**, which returns the vehicle ID number and transfers the vehicle's locations and movements from the `Record` class to a `Vehicle` class.
- The third function call is to **`getVehicleType`**, which returns a number for the evaporative vehicle type.

The 16 vehicle types are as follows:

- Pre 1980 vehicles, with carburetor, failed pressure test
- Pre 1980 vehicle, with carburetor, failed purge test only
- Pre 1980 vehicle, with carburetor, passed purge and pressure tests
- 1980-1985 vehicle, with carburetor, failed pressure test
- 1980-1985 vehicle, with carburetor, failed purge test only
- 1980-1985 vehicle, with carburetor, passed purge and pressure test
- 1980-1985 vehicle, fuel injected, failed pressure test
- 1980-1985 vehicle, fuel injected, failed purge test only
- 1980-1985 vehicle, fuel injected, passed both purge and pressure tests
- 1985-1998 vehicle, with carburetor, not certified to enhanced standards, failed pressure test
- 1985 –1998 vehicle, with carburetor, not certified to enhanced standards, failed purge test only
- 1985-1998 vehicle, with carburetor, not certified to enhanced standards, passed both pressure and purge tests
- 1985-1998 vehicle, fuel injected, not certified to enhanced standards, failed pressure test
- 1985-1998 vehicle, fuel injected, not certified to enhanced standards, failed purge test only
- 1985-1998 vehicle, fuel injected, not certified to enhanced standards, passed both pressure and purge tests

- Gross liquid leaker

The function **`getVehicleType`** does not identify if a vehicle is a gross liquid leaker or not, since the distribution for gross liquid leaders are different for the different types of evaporative emissions. The function **`getVehicleType`** calls functions to obtain the model year (**`getModelYear`**), determine whether the vehicle is a car or pickup (**`askCar`**), and to determine whether the vehicle has a carburetor or is fuel injected (**`askCarb`**).

These functions get the probabilities of being a particular vehicle from the `UserDefined` class and use a random number generator to determine the specific vehicle type. The last function call in **`getVehicleType`** is to **`defVehicleType`**, which calculates the probabilities of failing the pressure test and the probability of passing both the pressure and purge test from the EPA formulas. It then uses a random number to determine if it passed or failed the tests. This is combined with the vehicle model year to determine the evaporative vehicle type for the vehicle.

Once the vehicle type is determined, a call in *main* is made to the function **`askLeaker`**, which determines if the vehicle is a gross liquid leaker for the specific type of evaporative emission (resting loss, diurnal, or hot soak). The function **`askLeaker`** returns a value of 16 if the vehicle is a leaker; otherwise, it returns the original vehicle type number. The function uses EPA equations to calculate the probability of the vehicle being a gross liquid leaker and a random number to determine if the vehicle is a gross liquid leaker.

The resting loss evaporative emissions and their location are calculated for each 15-minute segment of the day. The function **`getRestLoss`** uses the following to calculate resting loss evaporative emissions:

- the evaporative vehicle type,
- EPA formulas,
- coefficients in the `Coefficient` class, and
- the hourly temperature in the `Temperature` class.

If the vehicle is a gross liquid leaker, the emissions are calculated in the function **`getLeaker`**. The emissions and their location are entered in an `Emission` class, which holds the total emissions for each 15-minute time period for all vehicles.

To calculate diurnal emissions, we begin by calling the function **`getDiurnal`** in *main*. The subroutine first calculates the 24-hour diurnal emissions and then finds what fraction of these emissions is emitted (depending of the vehicle's operating history).

The first section calculates the fraction of the 24-hour emissions that occur from the beginning of the simulation until the vehicle is operated. The EPA assumes that there are no diurnal emissions from midnight to 6 a.m., and thus in actuality the first step is to calculate the emissions from 6 a.m. until the vehicle is operated. The fractions of the 24-

hour emissions are calculated in the function **`getPartialDiurnal`**, which returns the fraction of the 24 diurnal emissions that should be allocated for the one-hour period.

The function **`getPartialDiurnal`** uses a different formula for each vehicle evaporation type to obtain the fraction of the 24-hour emissions that should be allocated to each hour. The next calculations of diurnal emission are those emissions that take place after a vehicle has been operating. The EPA assumes that diurnal emissions start two hours after a vehicle is turned off and continue as long as the ambient temperature is above the ambient temperature that existed two hours after the vehicle is turned off. Thus, if a vehicle is turned off in the late afternoon when temperatures are decreasing, there are no more diurnal emissions.

The code calculates emissions for the initial and final fractions of a 15-minute segment. If a vehicle has evaporative emissions in more than one location during a 15-minute segment, all emissions for that 15-minute segment are allocated to the last location of the vehicle. The diurnal emissions are added to the `Emission` class containing the total emission for a vehicle. The temporary `Emission` class then is zeroed out as in the resting loss calculation.

The last calculation in the non-operating evaporative emissions is the hot soak evaporative emission calculation. Because there is a different probability of a vehicle being a gross liquid leaker for hot soaks than for resting loss and diurnal emissions, the **`askLeaker`** function is called to determine if the vehicle is a gross liquid leaker.

The hot soak emissions are then calculated in the function **`getHotSoak`**. Hot soak emissions take place only one hour after a vehicle has been turned off or until the vehicle restarts (if that is less than one hour). Hot soak emissions are based on a different set of criteria than for the other evaporative emissions, so the function **`getHotsoak`** checks to see if the vehicle is a car or pickup and, if it is fuel injected, whether it uses a TBI or PFI fuel-injection system.

This is determined by using the percentage of vehicles that fit these categories given in the `UserDefined` class. Once the vehicle has been categorized, the hot soak emissions for one hour are calculated and assigned to the appropriate 15-minute segments. If the vehicle is restarted in less than one hour, the fractional hot soak is calculated and assigned to the appropriate 15-minute segment.

The 15-minute timesteps and hydrocarbon emissions for the duration of the simulation in the emissions object is output to the file *StationaryEvapEmiss.dat*. The emissions for a single vehicle is stored, and then the program loops to calculate the stationary emissions for the next vehicle. Once all of the stationary evaporative emissions are calculated, the program calculates the running loss emissions. The calculations are done in the `RunningLoss` class. The function **`getTotalRuningLoss`** takes hourly information on average speeds and vehicle occurrences for blocks along the network from the velocity summary data produced by the Traffic Microsimulator and the total city-wide trip length distribution for the time of day and calculates the running loss emissions for one hour for a block. The calculation uses the fractional distribution of vehicle ages, the fraction of vehicles that have carburetors, the fraction of cars and pickups, and the fraction of leakers along with the formulas in the EPA MOBILE5 model to calculate the hourly emissions for a block. This information is then output to a file in the same format as that for tailpipe emissions.

2.4 Usage

The Evaporative Emissions Estimator is designed to produce emissions appropriate for simulations of air quality over a metropolitan area. It is designed to produce resting loses, diurnal emissions, hot soak, and running losses. These losses are separated into operating and non-operating emissions output files.

Usage:

```
Evaporative Estimator <configFilename>
```

where

`<configFilename>` is the name of the configuration file used to run the Traffic Microsimulator to collect output

2.5 Inputs and Outputs

2.5.1 *evapCityData* File

The following information is read in from a city-specific data file *evapCityData* or the file specified by the `EMISSIONS_EVAP_CITY_FILE` configuration file key.

Input information is required in the `UserDefined` and `Temperature` classes. The `UserDefined` class needs information regarding

- the model year distribution of the vehicles,
- percentage of each model year that consists of trucks,
- percentage of vehicles that have a carburetor by model year,
- percentage of fuel injected vehicles by model year that have TBI systems,
- the present year of the run, and
- the RVP value for the fuel.

The `Temperature` class requires the hourly temperature distribution for the day.

For the stationary evaporative emissions, the model requires inputs from the traveler event output file containing

- the vehicle ID number,
- the location where the vehicle is turned on,

- the time of day when the vehicle was turned on (seconds from midnight),
- the location where the vehicle is turned off,
- the time of day when the vehicle was turned off (seconds from midnight), and
- the elapsed time of the vehicle trip (seconds).

2.5.2 *pa.out* File

The pa.out file (or file specified by the `EMISSIONS_PA_OUTPUT_FILE` configuration file key) contains the vehicle activity record information. This file is produced by the *ConvertTRVfile* program. The file contains records with the following fields: `vehicle id`, `start time`, `start parking location`, `end time`, `end parking location`, and `trip duration in seconds`.

If the vehicle is operating at the beginning or end of the simulation, a `-1` is input at the start or end time, respectively. This input file is assumed to contain the activities for all the light-duty vehicles sorted by the vehicle ID and the trip start time. See Appendix K for field descriptions..

2.5.3 *summary.ldv.vel* File

For the operating evaporative emission, the model requires the *summary.ldv.vel* file (or the file defined by the `EMISSIONS_MICROSIM_LDV_VELOCITY_FILE` configuration file key) that contains the counts per hour, per CA speed, per 30-meter link block.

2.5.4 *EMISSIONS_MATRICES.dat* File

This input file contains running loss emission coefficients for cars passing purge and pressure tests, vehicles failing purge and pressure tests, and trucks passing purge and pressure tests.

2.5.5 *StationaryEvapEmis.dat* File

The output of the first portion of the model is in the *StationaryEvapEmis.dat* file (or the file specified by `EMISSIONS_EVAP_STATIONARY_OUTFILENAME` configuration file key), which contains the total non-operative emissions for all vehicles found in the traveler event file. The output is a combination of 15-minute segment data into one-hour time slices. Its format is the similar to the tailpipe emission modules with only the HydroCarbon emissions outputted. See Appendix H for field descriptions.

2.5.6 *OperatingEvapEmis.dat* File

The running loss emissions are outputted in the file *OperatingEvapEmis.dat* (or the file specified by the `EMISSIONS_EVAP_OPERATING_OUTPUTFILENAME` configuration file key). The output format is similar to the tailpipe emissions with only the HydroCarbon emissions outputted.. See Appendix H for field descriptions.

2.6 Utility Programs

2.6.1 *ConvertTRVfile* Utility

The *ConvertTRVfile* program transforms the traveler event file into a file used as input into the Evaporative Emissions Estimator. The output file contains vehicle IDs, start time, start parking location ID, end time, end parking location ID, and trip length (in seconds) for all trips contained in the microsimulation traveler event file. The configuration file key `OUT_SUMMARY_EVENT_FILTER` should be set to

```
VEHTYPE==1;STATUS@[1286|16900|25860]
```

before the microsimulation is run so that only vehicles of type `AUTO` and only statuses of start and end trip are collected. This helps to reduce the size of the output file. Appendix K lists fields in the outputted vehicle activity file.

Appendix A: Configuration File Keys Specific to the Emissions Estimators

Note: Avoid using the percent (%) character when naming input and output files. This will cause problems with the logging system and output of warning and error messages.

Configuration File Key	Description
`EMISSION_RATIOS_LONG_SOAK_FILE`	Multipliers representing ratios of the different emissions by eight power levels for long soaks. Default = *longSoakRatios*

`EMISSIONS_ARRAY_PARAMETERS_FILE`
or
`EMISSIONS_HDV_ARRAY_PARAMETERS_FILE`
	The parameters describing the number of records and increments used in composite input files.
	Default = *ARRAYP.INP* or *ARRAY_HDV.INP*

`EMISSIONS_COMPOSITE_DIFF_INPUT_FILE`	The composite emissions for the differences in emissions and fuel consumption for current versus last timestep for road grades < 1% or downhill. Default = *arraypd.out*
`EMISSIONS_COMPOSITE_DIFF2P_INPUT_FILE`	The composite emissions for the differences for road grades between 1% and 3%. Default = *arraypd2p.out*
`EMISSIONS_COMPOSITE_DIFF4P_INPUT_FILE`	The composite emissions for the differences for road grades between 3% and 5%. Default = *arraypd4p.out*
`EMISSIONS_COMPOSITE_DIFF6P_INPUT_FILE`	The composite emissions for the differences for road grades above 5%. Default = *arraypd6p.out*
`EMISSIONS_COMPOSITE_HDV_INPUT_FILE`	The composite vehicle emissions in 4-mph speed bins and four power bins. Default = *catruck.acc*
`EMISSIONS_COMPOSITE_INPUT_FILE`	The composite vehicle emissions in 4-mph speed bins and 20-mph squared per second power bins for road grades < 1% or downhill. Default = *arrayp.out*
`EMISSIONS_COMPOSITE_TYPE_DIFF_INPUT_FILE`	The composite emissions for the difference in emissions and fuel consumption versus last timestep for 23 LDV subtypes. Default = *batchtotpd*
`EMISSIONS_COMPOSITE_TYPE_INPUT_FILE`	The composite emissions for 20 speeds, 34 power levels, and 23 LDV types. Default = *batchtotpc*
`EMISSIONS_COMPOSITE2P_INPUT_FILE`	The composite vehicle emissions or road grades between 1% and 3%. Default = *arrayp2p.out*

Configuration File Key	Description
EMISSIONS_COMPOSITE4P_INPUT_FILE	The composite vehicle emissions or road grades between 3% and 5%. Default = *arrayp4p.out*
EMISSIONS_COMPOSITE6P_INPUT_FILE	The composite vehicle emissions or road grades above 5%. Default = *arrayp6p.out*
EMISSIONS_DEBUG1_FILE or EMISSIONS_DEBUG1_HDV_FILE	First debugging file. Default = *debug.ldv.out* or *debug.hdv.out*
EMISSIONS_DEBUG2_FILE or EMISSIONS_DEBUG2_HDV_FILE	Second debugging file. Default = *calcsum.ldv* or *calcsum.hdv*
EMISSIONS_ENR_LONG_SOAK_FILE	The filename of the postprocessed energy file for the long soak vehicles. Created by the *ConvertENRfile* program. Default = *energy.long.out*
EMISSIONS_ENR_MEDIUM_SOAK_FILE	The filename of the postprocessed energy file for the medium soak vehicles. Created by the *ConvertENRfile* program. Default = *energy.medium.out*
EMISSIONS_ENR_NO_SOAK_FILE	The filename of the postprocessed energy file for the negligible soak vehicles. Created by the *ConvertENRfile* program. Default = *energy.no.out*
EMISSIONS_ENR_SHORT_SOAK_FILE	The filename of the postprocessed energy file for the short soak vehicles. Created by the *ConvertENRfile* program. Default = *energy.short.out*
EMISSIONS_EVAP_CITY_FILE	The name of the file containing city-specific data such as temperature distribution and vehicle data. Default = *evapCityData*
EMISSIONS_EVAP_COEF_FILE	The name of the file containing running loss coefficients. Default = *EMISSIONS_MATRICES.dat*
EMISSIONS_EVAP_DEBUG_FILENAME	The name of the Evaporative Emissions debugging file. Default = *debug.dat*
EMISSIONS_EVAP_OPERATING_OUTFILENAME	The name of the file into which operating loss emissions are written. Default = *OperatingEvapEmis.dat*
EMISSIONS_EVAP_STATIONARY_OUTFILENAME	The name of the file into which startionary loss emissions are written. Default = *StationaryEvapEmis.dat*
EMISSIONS_LDV_OUTPUT_FILE or EMISSIONS_HDV_OUTPUT_FILE	The filenames of the final output from the Tailpipe Emissions Estimator. Default = *emissions.ldv.out* or *emissions.hdv.out*

Configuration File Key	Description
EMISSIONS_LDV_VELOCITY_FILE or EMISSIONS_HDV_VELOCITY_FILE	The filename for the reformatted Traffic Microsimulator velocity output. These files are created by the ConvertVELfile utility program. Default = *velocity.ldv.out* or *velocity.hdv.out*
EMISSIONS_MICROSIM_ENR_LONG_SOAK_FILE	The filename of the Traffic Microsimulator energy file containing energies for long soak vehicles. Default = *summary.long.enr*
EMISSIONS_MICROSIM_ENR_MEDIUM_SOAK_FILE	The filename of the Traffic Microsimulator energy file containing energies for medium soak vehicles. Default = *summary.medium.enr*
EMISSIONS_MICROSIM_ENR_NO_SOAK_FILE	The filename of the Traffic Microsimulator energy file containing energies for negligible soak vehicles. Default = *summary.no.enr*
EMISSIONS_MICROSIM_ENR_SHORT_SOAK_FILE	The filename of the Traffic Microsimulator energy file containing energies for short soak vehicles. Default = *summary.short.enr*
EMISSIONS_MICROSIM_LDV_VELOCITY_FILE or EMISSIONS_MICROSIM_HDV_VELOCITY_FILE	The filenames for the Traffic Microsimulator velocity output data. Default = *summary.ldv.vel* or *summary.hdv.vel*
EMISSIONS_MICROSIM_TRAVELER_FILE	The filename of the Traffic Microsimulator traveler event file. Default = *event.trv*
EMISSIONS_PA_OUTPUT_FILE	The filename of the parking output file from *ConvertTRVfile*. Default = *pa.out*
EMISSIONS_RATIOS_MEDIUM_SOAK_FILE	Multipliers representing ratios of the different emissions by eight power levels for medium soaks. Default = *mediumSoakRatios*
EMISSIONS_RATIOS_SHORT_SOAK_FILE	Multipliers representing ratios of the different emissions by eight power levels for short soaks. Default = *shortSoakRatios*
EMISSIONS_SUBTYPE_OUTPUT_FILE	The filename of the vehicle subtype output file from *ConvertTRVfile*. Default = *sub.out*
EMISSIONS_VEHICLE_TYPE_DISTRIBUTION	The distributions by 23 LDV types. Default = *vehdist*
EMISSIONS_WRITE_DEBUG_OUTPUT	Whether to create the two debug files or not Default = 0 (not to write out)

Appendix B: Configuration File Keys That Must Be Set to a Specific Value for the Emissions Estimators

Configuration File Key	Description
CA_CELL_LENGTH	The length of a cell that a vehicle occupies (in meters). Must be set to 7.5.
CA_LONG_SOAK_TIME	Time where medium vs. long soak is determined (in seconds). Must be set to 9000.
CA_MEDIUM_SOAK_TIME	Time where short vs. medium soak is determined (in seconds). Must be set to 1800.
CA_SHORT_SOAK_TIME	Time where negligible vs. short soak is determined (in seconds). Must be set to 600.
NET_ACTIVITY_LOCATION_TABLE	The activity location table name.
NET_DIRECTORY	The full path name to the directory containing the network tables.
NET_LINK_TABLE	The name of the link table.
NET_NODE_TABLE	The name of network's node table.
NET_PARKING_TABLE	The parking table name.
NET_PROCESS_LINK_TABLE	The process link table name.
NET_TRANSIT_STOP_TABLE	The transit stop table name.
OUT_SUMMARY_BOX_LENGTH_n or OUT_SUMMARY_BOX_LENGTH_DEFAULT	The length of the roadway used to collect summary data (in meters). Must be set to 30.
OUT_SUMMARY_ENERGY_BINS_n or OUT_SUMMARY_ENERGY_BINS_DEFAULT	The number of bins to cover the range of energy histograms. Must be set to 7.
OUT_SUMMARY_ENERGY_MAX_n or OUT_SUMMARY_ENERGY_MAX_DEFAULT	The maximum energy for the range of energies found in energy histograms. Must be set to 105.
OUT_SUMMARY_ENERGY_SOAK_n	The type of energy soak data to collect. There must be four file specifications (one for each soak type): NEGLIGIBLE, SHORT, MEDIUM, and LONG.
OUT_SUMMARY_SAMPLE_TIME_n or OUT_SUMMARY_SAMPLE_TIME_DEFAULT	The frequency (in seconds) at which to accumulate velocity data. Must be set to 1.
OUT_SUMMARY_TIME_STEP_n or OUT_SUMMARY_TIME_STEP_DEFAULT	The frequency (in seconds) at which to report velocity and energy data. Must be set to 3600 (except for summary time data set to 900).
OUT_SUMMARY_TYPE_n	The type of summary output to collect. Specifications are needed for VELOCITY and ENERGY.

Configuration File Key	Description
`OUT_SUMMARY_VEHICLE_TYPE_n`	The type of velocity summary data to collect. Set to `AUTO` to collect LDV data, and either `TRUCK` or `BUS` for HDV data.

`OUT_SUMMARY_VELOCITY_BINS_n`
or
`OUT_SUMMARY_VELOCITY_BINS_DEFAULT`
	The number of bins used to cover the range of the velocity histogram. Must be set to 5.

`OUT_SUMMARY_VELOCITY_MAX_n`
or
`OUT_SUMMARY_VELOCITY_MAX_DEFAULT`
	The maximum velocity for range of velocities found in velocity histograms. Must be set to 37.5.

Appendix C-1: Fields in the Array Parameters File

Field	Description
`T0`	The time since engine start; not used.
`RGRADE0`	The representative minimum grade; not used.
`DRGRADE`	The spacing in grade arrays; not used.
`V0ARRAY`	The representative speed for the lowest speed index (mph); not used.
`DVARRAY`	The speed bin size (mph).
`A0ARRAY`	The representative power for the lowest power index (mph squared per sec).
`DACCARAY`	The acceleration bin size (mph**2/sec).
`NGRADE`	The number of grades in the emission arrays; not used.
`NVARRY`	The number of velocity bins in the emission arrays.
`NAARRAY`	The number of power bins in the emission arrays.

Appendix C-2: Fields in the Eight Composite Files

Field	Description
`VARRAY`	The representative speed (mph) for emissions calculation; not used.
`ACARRAY`	The representative acceleration for emissions calculation; not used.
`HCTIJK`	The hydrocarbon tailpipe emission rate (grams/sec).
`COTIJK`	The carbon monoxide tailpipe emission rate (grams/sec).
`NOXTIJK`	The nitrogen oxides tailpipe emission rate (grams/sec).
`FECON`	The fuel consumption rate (grams/sec).

Appendix D: Fields in the *SoakRatio Files

Field	Description
`HCR [soak] [power]`	The multiplier for hydrocarbons for a particular power level (eight of them)
`COR [soak] [power]`	The multiplier for carbon monoxide for a particular power level (eight of them)
`XNOXR [soak][power]`	The multiplier for nitrogen oxide for a particular power level (eight of them)
`FCR [soak] [power]`	The multiplier for fuel economy for a particular power level (eight of them)

Appendix E: Fields in the Link Velocity Files

velocity.ldv.out and *velocity.hdv.out* (assuming the Traffic Microsimulator was running with `OUT_SUMMARY_VELOCITY_BINS` set to "`5`")

Field	Description
`NV`	The number of velocity records for this link, equivalent to the number of boxes that partition the link.
`TIME`	The current time (seconds from midnight)
`LINK`	The link ID being reported.
`NODE`	The node ID from which the vehicles were traveling away.
`DISTANCE1`	The ending distance, from which the setback of the node from which vehicles were traveling away, of the 1st box with data it in.
`DISTANCE2`	The ending distance, from the setback of the node from which vehicles were traveling away, of the last box with data in it.
`COUNT0`	The number of vehicles with velocities in the range [0, 7.5).
`COUNT1`	The number of vehicles with velocities in the range [7.5, 15).
`COUNT2`	The number of vehicles with velocities in the range [15, 22.5).
`COUNT3`	The number of vehicles with velocities in the range [22.5, 30).
`COUNT4`	The number of vehicles with velocities in the range [30, 37.5).
`COUNT5`	The number of vehicles with velocities in the range [37.5, infinity).

Appendix F: Fields in the Link Energy Files

energy.no.out, energy.short.out, energy.med.out, and *energy.long.out* (assuming the Traffic Microsimulator was running with `OUT_SUMMARY_ENERGY_BINS` set to 7)

Field	Description
`TIME`	The current time (seconds from midnight).
`LINK`	The link ID being reported.
`NODE`	The node ID from which the vehicles were traveling away.
`ENERGY0`	The fraction of vehicles whose energy was in the lowest energy bin and a soak time associated with the particular energy soak file.
`ENERGY1`	The fraction of vehicles whose energy was in the second energy bin and a soak time associated with the particular energy soak file.
`ENERGY2`	The fraction of vehicles whose energy was in the third energy bin and a soak time associated with the particular energy soak file.
`ENERGY3`	The fraction of vehicles whose energy was in the fourth energy bin and a soak time associated with the particular energy soak file.
`ENERGY4`	The fraction of vehicles whose energy was in the fifth energy bin and a soak time associated with the particular energy soak file.
`ENERGY5`	The fraction of vehicles whose energy was in the sixth energy bin and a soak time associated with the particular energy soak file.
`ENERGY6`	The fraction of vehicles whose energy was in the seventh energy bin and a soak time associated with the particular energy soak file.
`ENERGY7`	The fraction of vehicles whose energy was in the highest energy bin and a soak time associated with the particular energy soak file.

Appendix G: Tailpipe Emissions Output for the Output Visualizer

for *emissions.ldv.out* and *emissions.hdv.out*

Field	Description
TIME	The current time (seconds from midnight).
LINK	The link ID being reported.
NODE	The node ID vehicles were traveling away from.
DISTANCE	The ending distance of the box (in meters) from the setback of the node from which the vehicles were traveling away.
LENGTH	The length of box.
VTT	The average speed in feet per second.
NOX	The nitrogen oxides emissions (milligrams per 30-meter segment).
CO	The carbon monoxide emissions (grams per 30-meter segment).
HC	The hydrocarbon emissions (milligrams per 30-meter segment).
FE	The fuel consumption (grams per 30-meter segment).
FLUX	The vehicle flux in number times speed in feet per second.

Appendix H: Evaporative Emissions Output for the Output Visualizer

for *StationaryEvapEmis.dat* and *OperatingEvapEmis.dat*

Field	Description
`TIME`	The current time (seconds from midnight).
`LINK`	The link ID being reported.
`NODE`	The node ID vehicles were traveling away from.
`DISTANCE`	The ending distance of the box (in meters) from the setback of the node from which the vehicles were traveling away.
`LENGTH`	The length of box.
`HC`	The hydrocarbon emissions (milligrams per 30-meter segment).

Appendix I: Fields in the *debug.*.out* Debug Output Files

Field	Description
`ICX`	The segment of which the calculations are made. This includes the intersection.
`DELTAF`	The width of the highest speed bin (always 24.6 feet per second).
`COUNTS`	The number of vehicles at different velocities for this segment.
`DEN`	The fitted average number of vehicles per 7.5-meter cell for the six velocity bins.
`FIJ`	The estimated average vehicle densities per spatial cell (24.6 feet) and per speed cell (24.6 feet per second) for the six velocity bins.
`HIJ`	The gradient in estimated average vehicle density in units of number per spatial cell squared per speed cell for the six velocity bins.
`VEHFLUX`	The estimated vehicle flux in each speed bin for speed bins 0-5 in units of number times feet per second for the six velocity bins
`VEHFT`	The total estimated vehicle flux in speed bins per cell.
`VEHD`	The estimated number of vehicles in each speed bin in each cell for the six velocity bins.
`VEHDT`	The estimated total number of vehicles in a spatial cell.
`VBAR`	The estimated mean speed in feet per second.
`SDEVRAT`	The estimated ratio of the standard deviation of speed to mean speed.
`VLOWRI`	The cutoff speed for the slowest one-third of the vehicles defined by flux in feet per second.
`VUPPRI`	The cutoff speed for the slowest two-thirds of the vehicles defined by flux in feet per second.
`V2SDEV`	The product of the square of the mean speed and the difference between the speed standard deviation and its low-congestion reference value in units of feet cubed per second cubed.
`ITAR`	The distance index for the lower fluxes.
`IREF`	The distance index for the higher fluxes.
`SPDCL`	The estimated gradient in the cube of the speed normalized by the cube of a spatial cell per second (24.6**3) in units of inverse feet for the slowest third of the flux.
`SPDCM`	The estimated gradient in the cube of the speed normalized by the cube of a spatial cell per second (24.6**3) in units of inverse feet for the middle one-third of the flux.
`SPDCH`	The estimated gradient in the cube of the speed normalized by the cube of a spatial cell per second (24.6**3) in units of inverse feet for the fastest one-third of the flux.
`VEHDL`	The average vehicle density for the slowest one-third of vehicles.
`VEHDM`	The average vehicle density for the middle one-third of vehicles.
`VEHDH`	The average vehicle density for the fastest one-third of vehicles.
`VEHFLUXL`	The estimated vehicle flux for the slowest one-third of the vehicles for the current segment, followed by that of the next four segments down the link.
`VEHFLUXM`	The estimated vehicle flux for the middle one-third of the vehicles for the current segment, followed by that of the next four segments down the link.
`VEHFLUXH`	The estimated vehicle flux for the fastest one-third of the vehicles for the current segment, followed by that of the next four segments down the link.

Field	Description
VCUBEDL	The estimated average cube of the velocity in units of feet cubed per second cubed for the slowest one-third of the vehicle for the current segment followed by that of four following segments down the link.
VCUBEDM	The estimated average cube of the velocity in units of feet cubed per second cubed for the middle one-third of the vehicle for the current segment followed by that of four following segments down the link.
VCUBEDH	The estimated average cube of the velocity in units of feet cubed per second cubed for the fastest one-third of the vehicle for the current segment followed by that of four following segments down the link.
PDL	The fraction of the vehicles in the slowest one-third of the flux that undergo a hard deceleration.
PL	The probability of a hard acceleration in the slowest one-third.
PNS	The fraction of vehicles in the one- third under consideration that have insignificant acceleration or deceleration.
PTOTFL	The total power in the one-third that is under consideration.
PTOTMI	The power in the hard-decelerating vehicles in the one-third that is under consideration.
PTOTPP	The power in the high-power vehicles in the one-third under consideration.
PPNS	The average power of vehicles in the insignificant power category.
PDC	The fraction of vehicles in the middle one-third of the flux that undergo a hard deceleration.
PCC	The probability of a hard deceleration in the middle one-third.
PDH	The fraction of vehicles in the fastest one-third of the flux that undergo a hard deceleration.
PH	The probability of a hard acceleration in the fastest one-third.
ICX	The segment for which the output is reported.
XNOSUL	The estimated NO_x emissions for the slowest one-third in units of grams per 7.5-meter cell.
XNOSUC	The estimated NO_x emissions for the middle one-third in unites of grams per 7.5-meter cell.
XNOSUH	The estimated NO_x emissions for the fastest one-third in units of grams per 7.5-meter cell.
COSUL	The estimated CO emissions for the slowest one-third in units of grams per 7.5-meter cell.
COSUC	The estimated CO emissions for the middle one-third in units of grams per 7.5-meter cell.
COSUH	The estimated CO emissions for the fastest one-third in units of grams per 7.5-meter cell.
V2SDEV	The product of the square of the mean speed and the difference between the speed standard deviation and its low congestion reference value in units of feet cubed per second cubed.
SDEV	The standard deviation of speed derived from the estimated distribution.
PL	The probability of a hard acceleration in the slowest one-third; unlike the earlier reference, this includes an adjustment if the slowest one-third is in the first speed bin.
PCC	The probability of a hard acceleration in the middle one-third; unlike the earlier reference, this includes an adjustment if the middle one-third is in the first speed bin.

Field	Description
`PH`	The probability of a hard acceleration in the fastest one-third; unlike the earlier reference, this includes an adjustment if the fastest one-third is in the first speed bin.

Appendix J: Fields in the *calcsum** Debugging Output Files

Field	Description
ICX	The segment for which output is reported.
DSUMLF	The power of hard-decelerating vehicles in the slowest one-third of the flux.
ZSUMLF	The power of insignificant-power vehicles in the slowest one-third of the flux.
ASUMLF	The power of high-power vehicles in the slowest one-third of the flux.
DSUMCF	The power of hard-decelerating vehicles in the middle one-third of the flux.
ZSUMCF	The power of insignificant-power vehicles in the middle one-third of the flux.
ASUMCF	The power of the high-power vehicles in the middle one-third of the flux.
DSUMHF	The power of the hard-decelerating vehicles in the fastest one-third of the flux.
ZSUMHF	The power of the insignificant-power vehicles in the fastest one-third of the flux.
ASUMHF	The power of the high-power vehicles in the fastest one-third of the flux.
PDL	The fraction of the vehicles in the slowest one-third of the flux that undergo a hard deceleration.
PL	The probability of a hard acceleration in the slowest one-third.
PDC	The fraction of vehicles in the middle one-third of the flux that undergo a hard deceleration.
PCC	The probability of a hard acceleration in the middle one-third.
PDH	The fraction of vehicles in the fastest one-third of the flux that undergo a hard deceleration.
PH	The probability of a hard acceleration in the fastest one-third.
V2SDEV	The product of the square of the mean speed and the difference between the speed standard deviation and its low congestion reference value in units of feet cubed per second cubed.
SPDLI	The speed parameter (spdc) for the slowest one-third of the flux, smoothed over three segments.
SPDMI	The speed parameter (spdc) for the middle one-third of the flux, smoothed over three segments.
SPDHI	The speed parameter (spdc) for the fastest one-third of the flux, smoothed over three segments.
SDEV	The standard deviation for speed derived from estimated distribution.
VEHDL	The average density for the slowest one-third of vehicles per 7.5-meter cell.
VEHDM	The average density for the middle one-third of vehicles per 7.5-meter cell.
VEHDH	The average density for the fastest one-third of vehicles per 7.5-meter cell.
SPDCT	The speed parameter (spdc) for the entire flux.

Appendix K: Fields in the Vehicle Type Distribution Files

Field	Description
`FRACTION1`	The fraction of LDVs of subtype1.
`FRACTION2`	The fraction of LDVs of subtype2.
`FRACTION3`	The fraction of LDVs of subtype3.
`FRACTION4`	The fraction of LDVs of subtype4.
`FRACTION5`	The fraction of LDVs of subtype5.
`FRACTION6`	The fraction of LDVs of subtype6.
`FRACTION7`	The fraction of LDVs of subtype7.
`FRACTION8`	The fraction of LDVs of subtype8.
`FRACTION9`	The fraction of LDVs of subtype9.
`FRACTION10`	The fraction of LDVs of subtype10.
`FRACTION11`	The fraction of LDVs of subtype11.
`FRACTION12`	The fraction of LDVs of subtype12.
`FRACTION13`	The fraction of LDVs of subtype13.
`FRACTION14`	The fraction of LDVs of subtype14.
`FRACTION15`	The fraction of LDVs of subtype15.
`FRACTION16`	The fraction of LDVs of subtype16.
`FRACTION17`	The fraction of LDVs of subtype17.
`FRACTION18`	The fraction of LDVs of subtype18.
`FRACTION19`	The fraction of LDVs of subtype19.
`FRACTION20`	The fraction of LDVs of subtype20
`FRACTION21`	The fraction of LDVs of subtype21.
`FRACTION22`	The fraction of LDVs of subtype22.
`FRACTION23`	The fraction of LDVs of subtype23.

Appendix L: Fields in the Traveler Event Postprocessed File (*pa.out*)

Field	Description
`VEHICLE`	The vehicle ID of vehicle for this record.
`START_PA`	The parking location ID from which the vehicle started this particular trip.
`START_TIME`	The time (in seconds since midnight) that the vehicle began this particular trip at the `START_PA` (-1 if the vehicle started before the simulation start-time).
`END_PA`	The parking location ID at which the vehicle ended this particular trip (-1 if simulation ended before the trip ended).
`END_TIME`	The time (in seconds since midnight) that the vehicle ended this particular trip at the `END_PA` (-1 if simulation ended before the trip ended).
`VEHSEC`	The number of seconds the vehicle spent on this particular trip (`END_TIME - START_TIME`) [-1 if the trip began before simulation start time or if simulation ended before the trip ended).

Appendix M: Emissions Estimators' Library Files

Type	File Name	Description
Binary Files	*libTIO.a*	The TRANSIMS Interfaces library.
	libGlobals.a	The TRANSIMS Global library.
	libNetwork.a	The TRANSIMS Network library.
	libError.a	The TRANSIMS Error library.
Source Files	*emissionsEstimator.C*	The main LDV emissions module that takes Traffic Microsimulator velocity summary data and outputs emissions that can be displayed in the Output Visualizer.
	emissionsEstimatorHDV.C	the main HDV emissions module that takes Traffic Microsimulator velocity summary data and outputs emissions that can be displayed in the Output Visualizer.
	ENVConfigKeys.h	The emissions configuration file keys.
	ENVError.[Cth]	The emissions error code files.
	emissions.h	The emissions print utilities.
	convertVELfile.C	Reads in a Traffic Microsimulator velocity summary output file and outputs the velocity data in a format that can be inputted to the Tailpipe Emissions Estimator.
	convertENRfile.C	Reads in four Traffic Microsimulator energy files and converts the counts into ratios over all four soak files.
	convertTRVfile.C	Reads in a Traffic Microsimulator traveler event file and creates an output file (used in the Evaporative Emissions Estimator) that contains individual vehicle trip start and stop locations and time information.
	CreateComposites.C	Converts the composite arrays by vehicle subtype into composite arrays with only one vehicle type. *batchtotpc* → *arrayp.out* *batchtotpd* → *arraypd.out*
	distribVELfile.C	Distributes postprocessed velocity output into several velocity files.
	combineEmissions.C	Combines several emissions output files into a single file (used when the *EmissionsEstimator* is run with distributed velocity files).
	Coefficient.C	Coefficient constructor loads coef arrays (hard coded values) and accessors.
	Coefficient.h	Definitions for `Coefficient` class.
	CreateNetwork.C	Functions for creating the network and converting a ParkingId to Link, Node, Distance, and Length.
	Emission.C	`Emission` constructor and accessors. Holds emissions as calculated for up to 48-hour 15-minute time slices.
	Emission.h	Definitions for `Emission` class
	MyFunct.h	Definitions for functions used in more than one file.
	Record.C	`Record` constructor and accessors. Holds individual vehicle activity data read in from the vehicle activity file.
	Record.h	Definitions for `Record` class
	RunningLoss.C	`RunningLoss` constructor and accessors. Used to calculate running losses.

Type	File Name	Description
	RunningLoss.h	Definitions for `RunningLoss` class.
	RunningLossCoef.C	`RunningLossCoef` constructor and accessors. Holds emissions coefficients.
	RunningLossCoef.h	Definitions for `RunningLossCoef` class.
	SimInfo.C	`SimInfo` constructor and accessors. Holds data read from velocity file and overall stats such as total trips and trip lengths.
	SimInfo.h	Definitions for `SimInfo` class.
	Temperature.C	`Temperature` constructor and accessors.
	Temperature.h	Definitions for `Temperature` class.
	UserDefined.C	`UserDefined` constructor and accessors.
	UserDefined.h	Definitions for `UserDefined` class.
	VaporPressure.C	`VaporPressure` constructor and accessors.
	VaporPressure.h	Definitions for `VaporPressure` class.
	Vehicle.C	`Vehicle` constructor and accessors. Holds postprocessed vehicle activity data.
	Vehicle.h	Definitions for X class.
	WriteRunningEmissions.C	Outputs running loss emissions.
	askCar.C	Function determines whether the vehicle is a car of a pickup.
	askCarb.C	Function determines whether a vehicle is carburated or is fuel injected.
	askLeaker.C	Function determines if a vehicle is a gross leaker for the different evaporation categories.
	combineEvap.C	Combines the Stationary and Operating Evaporative emissions files.
	defVehicleType.C	Function determines the evaporative emission type for a vehicle.
	evaporative.C	Main evaporative emissions module.
	evaporative.h	Global constants for evaporative module.
	getDirunal.C	Function calculates the 24-hour, partial, and interrupted diurnal emissions.
	getHotSoak.C	Function calculates the hot soak emissions.
	getLeaker.C	Function calculates the resting losses for leakers.
	getModelYear.C	Function to determine the model year of a vehicle.
	getPartialDiurnal.C	Function calculates the ratio of the hourly diurnal emissions to the total diurnal emissions for 24 hours.
	getRestLoss.C	Function calculates the resting loss evaporative emissions.
	getVehicleActivity.C	Function fills in a vehicle object with the vehicle activity information read in.
	getVehicletype.C	Function to set vehicle model year, whether a car or truck and whether carburated.
	readCityData.C	Function reads in city specific data regarding temperature distributions and vehicle information.
	readPAfile.C	Function reads the vehicle activity file created by *ConvertTRVfile*.

Appendix N: Examples

<u>Example 1</u> presents some of the configuration parameters that pertain to the Emissions Estimator.

Example 1. Configuration parameters.

```
TRANSIMS_ROOT                                /YourLocalDirectory

OUT_DIRECTORY                                $TRANSIMS_ROOT/output

NET_DIRECTORY                                $TRANSIMS_ROOT/data/calibration/tee/network
NET_NODE_TABLE                               Calibration_2_Nodes
NET_LINK_TABLE                               Calibration_2_Links
NET_PARKING_TABLE                            Calibration_2_Parking
NET_ACTIVITY_LOCATION_TABLE                  Calibration_2_Activities
NET_TRANSIT_STOP_TABLE                       Calibration_2_Transit
NET_PROCESS_LINK_TABLE                       Calibration_2_Process

CA_SIM_START_SECOND                          0
CA_SIM_START_MINUTE                          0
CA_SIM_START_HOUR                            0
CA_SIM_STEPS                                 86400

CA_SHORT_SOAK_TIME                           600     # 10 minutes
CA_MEDIUM_SOAK_TIME                          1800    # 30 minutes
CA_LONG_SOAK_TIME                            9000    # 2.5 hours

OUT_BEGIN_TIME_DEFAULT                       0
OUT_END_TIME_DEFAULT                         200000

OUT_EVENT_NAME_1                             event
OUT_EVENT_TYPE_1                             TRAVELER
OUT_EVENT_FILTER                             VEHTYPE==1;STATUS@[1286|16900|25860]

OUT_LINKS_DEFAULT                            $TRANSIMS_ROOT/data/output-
specs/tee_output_links
OUT_SUMMARY_TIME_STEP_DEFAULT                3600
OUT_SUMMARY_SAMPLE_TIME_DEFAULT              1
OUT_SUMMARY_BOX_LENGTH_DEFAULT               30

OUT_SUMMARY_ENERGY_BINS_DEFAULT              7
OUT_SUMMARY_ENERGY_MAX_DEFAULT               105

OUT_SUMMARY_NAME_1                           summary.no
OUT_SUMMARY_TYPE_1                           ENERGY
OUT_SUMMARY_ENERGY_SOAK_1                    NEGLIGIBLE
OUT_SUMMARY_ENERGY_BINS_1                    0
OUT_SUMMARY_ENERGY_MAX_1                     0

OUT_SUMMARY_NAME_2                           summary.short
OUT_SUMMARY_TYPE_2                           ENERGY
OUT_SUMMARY_ENERGY_SOAK_2                    SHORT

OUT_SUMMARY_NAME_3                           summary.medium
OUT_SUMMARY_TYPE_3                           ENERGY
OUT_SUMMARY_ENERGY_SOAK_3                    MEDIUM

OUT_SUMMARY_NAME_4                           summary.long
OUT_SUMMARY_TYPE_4                           ENERGY
OUT_SUMMARY_ENERGY_SOAK_4                    LONG

OUT_SUMMARY_VELOCITY_BINS_DEFAULT            5
OUT_SUMMARY_VELOCITY_MAX_DEFAULT             37.5

OUT_SUMMARY_NAME_5                           summary.auto
OUT_SUMMARY_TYPE_5                           VELOCITY
OUT_SUMMARY_VEHICLE_TYPE_5                   AUTO
```

```
OUT_SUMMARY_NAME_6                         summary.bus
OUT_SUMMARY_TYPE_6                         VELOCITY
OUT_SUMMARY_VEHICLE_TYPE_6                 BUS

LOG_EMISSIONS                              1
EMISSIONS_WRITE_DEBUG_OUTPUT               0
EMISSIONS_DEBUG1_FILE                      $TRANSIMS_ROOT/output/debug.ldv.out
EMISSIONS_DEBUG2_FILE                      $TRANSIMS_ROOT/output/calcsum.ldv
EMISSIONS_DEBUG1_HDV_FILE                  $TRANSIMS_ROOT/output/debug.hdv.out
EMISSIONS_DEBUG2_HDV_FILE                  $TRANSIMS_ROOT/output/calcsum.hdv

EMISSIONS_LDV_OUTPUT_FILE                  $TRANSIMS_ROOT/output/emissions.auto.out
EMISSIONS_HDV_OUTPUT_FILE                  $TRANSIMS_ROOT/output/emissions.bus.out

EMISSIONS_MICROSIM_LDV_VELOCITY_FILE       $TRANSIMS_ROOT/output/summary.auto.vel
EMISSIONS_MICROSIM_HDV_VELOCITY_FILE       $TRANSIMS_ROOT/output/summary.bus.vel

EMISSIONS_LDV_VELOCITY_FILE                $TRANSIMS_ROOT/output/velocity.auto.out
EMISSIONS_HDV_VELOCITY_FILE                $TRANSIMS_ROOT/output/velocity.bus.out

EMISSIONS_ARRAY_PARAMETERS_FILE            $TRANSIMS_ROOT/data/emissions/ARRAYP.INP
EMISSIONS_HDV_ARRAY_PARAMETERS_FILE        $TRANSIMS_ROOT/data/emissions/ARRAYP_HDV.INP

EMISSIONS_COMPOSITE_INPUT_FILE             $TRANSIMS_ROOT/data/emissions/arrayp.out
EMISSIONS_COMPOSITE2P_INPUT_FILE           $TRANSIMS_ROOT/data/emissions/arrayp2p.out
EMISSIONS_COMPOSITE4P_INPUT_FILE           $TRANSIMS_ROOT/data/emissions/arrayp4p.out
EMISSIONS_COMPOSITE6P_INPUT_FILE           $TRANSIMS_ROOT/data/emissions/arrayp6p.out

EMISSIONS_COMPOSITE_DIFF_INPUT_FILE        $TRANSIMS_ROOT/data/emissions/arraypd.out
EMISSIONS_COMPOSITE_DIFF2P_INPUT_FILE      $TRANSIMS_ROOT/data/emissions/arraypd2p.out
EMISSIONS_COMPOSITE_DIFF4P_INPUT_FILE      $TRANSIMS_ROOT/data/emissions/arraypd4p.out
EMISSIONS_COMPOSITE_DIFF6P_INPUT_FILE      $TRANSIMS_ROOT/data/emissions/arraypd6p.out

EMISSIONS_COMPOSITE_HDV_INPUT_FILE         $TRANSIMS_ROOT/data/emissions/catruck.accb

EMISSIONS_MICROSIM_ENR_NO_SOAK_FILE        $TRANSIMS_ROOT/output/summary.no.enr
EMISSIONS_MICROSIM_ENR_SHORT_SOAK_FILE     $TRANSIMS_ROOT/output/summary.short.enr
EMISSIONS_MICROSIM_ENR_MEDIUM_SOAK_FILE    $TRANSIMS_ROOT/output/summary.medium.enr
EMISSIONS_MICROSIM_ENR_LONG_SOAK_FILE      $TRANSIMS_ROOT/output/summary.long.enr

EMISSIONS_ENR_NO_SOAK_FILE                 $TRANSIMS_ROOT/output/energy.no.out
EMISSIONS_ENR_SHORT_SOAK_FILE              $TRANSIMS_ROOT/output/energy.short.out
EMISSIONS_ENR_MEDIUM_SOAK_FILE             $TRANSIMS_ROOT/output/energy.med.out
EMISSIONS_ENR_LONG_SOAK_FILE               $TRANSIMS_ROOT/output/energy.long.out

EMISSIONS_RATIOS_SHORT_SOAK_FILE           $TRANSIMS_ROOT/data/emissions/shortSoakRatios
EMISSIONS_RATIOS_MEDIUM_SOAK_FILE          $TRANSIMS_ROOT/data/emissions/mediumSoakRatios
EMISSIONS_RATIOS_LONG_SOAK_FILE            $TRANSIMS_ROOT/data/emissions/longSoakRatios

EMISSIONS_MICROSIM_TRAVELER_FILE           $TRANSIMS_ROOT/output/event.trv
EMISSIONS_PA_OUTPUT_FILE                   $TRANSIMS_ROOT/output/pa.out

EMISSIONS_COMPOSITE_TYPE_INPUT_FILE        $TRANSIMS_ROOT/data/emissions/batchtotpc
EMISSIONS_COMPOSITE_TYPE_DIFF_INPUT_FILE   $TRANSIMS_ROOT/data/emissions/batchtotpd
EMISSIONS_VEHICLE_TYPE_DISTRIBUTION        $TRANSIMS_ROOT/data/emissions/vehhist

EMISSIONS_EVAP_DEBUG_FILENAME              $TRANSIMS_ROOT/debug.dat
EMISSIONS_EVAP_COEF_FILE                   $TRANSIMS_ROOT/data/emissions/EMISSIONS_MATRICES.dat
EMISSIONS_EVAP_CITY_FILE                   $TRANSIMS_ROOT/data/emissions/evapCityData.Portland
EMISSIONS_EVAP_STATIONARY_OUTFILENAME      $TRANSIMS_ROOT/StationaryEvapEmis.dat
EMISSIONS_EVAP_OPERATING_OUTFILENAME       $TRANSIMS_ROOT/OperatingEvapEmis.dat
ENV_RANDOM_SEED                            #
```

<u>Example 2</u> shows a portion of a Traffic Microsimulator velocity summary file. These data were collected on the intersection calibration network by using the configuration parameters set to the values in Example 2. The data consist of the velocity bins for link 1, starting at node 6 at timestep 3600. There are seventeen boxes on that particular link, seven of which are never entered. Notice that the last box is only 15 meters long instead of 30 meters.

Example 2. Velocity summary file outputted from the Traffic Microsimulator.

COUNT0	COUNT1	COUNT2	COUNT3	COUNT4	COUNT5	DISTANCE	LINK	NODE	TIME
0	0	0	0	0	0	30	1	6	3600
0	0	0	0	0	0	60	1	6	3600
0	0	0	0	0	0	90	1	6	3600
0	0	0	0	0	0	120	1	6	3600
0	0	0	0	0	0	150	1	1	3600
0	0	0	0	0	0	180	1	6	3600
0	0	0	0	0	0	210	1	6	3600
3968	0	0	0	0	0	240	1	6	3600
16381	5346	551	218	722	0	270	1	6	3600
17431	5086	764	293	675	0	300	1	6	3600
17640	5015	827	268	679	0	330	1	6	3600
17733	5084	791	314	651	0	360	1	6	3600
17985	5135	802	274	665	0	390	1	6	3600
18042	5099	857	259	654	0	420	1	6	3600
18083	5141	835	288	620	0	450	1	6	3600
18157	5328	807	299	605	0	480	1	6	3600
9156	3107	358	184	234	0	495	1	6	3600

Example 3 shows a portion of a *velocity.ldv.out* file. The *velocity.ldv.out* file is created by the *ConvertVELfile* program, which reformats the Traffic Microsimulator output into a format that can be read in by the Tailpipe Emissions Estimator. Example 3 contains the output from the sample data in Example 2. Notice that the counts for the last box have been multiplied by 2. This is done in order to convert the 15-meter segment into a 30-meter segment.

Example 3. *velocity.ldv.out* file.

nv=	10	3600.0	1	6	240.0	495.0
	3.9680E+03	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	1.6381E+04	5.3460E+03	5.5100E+02	2.1800E+02	7.2200E+02	0.0000E+00
	1.7431E+04	5.0860E+03	7.6400E+02	2.9300E+02	6.7500E+02	0.0000E+00
	1.7640E+04	5.0150E+03	8.2700E+02	2.6800E+02	6.7900E+02	0.0000E+00
	1.7733E+04	5.0840E+03	7.9100E+02	3.1400E+02	6.5100E+02	0.0000E+00
	1.7985E+04	5.1350E+03	8.0200E+02	2.7400E+02	6.6500E+02	0.0000E+00
	1.8042E+04	5.0990E+03	8.5700E+02	2.5900E+02	6.5400E+02	0.0000E+00
	1.8083E+04	5.1410E+03	8.3500E+02	2.8800E+02	6.2000E+02	0.0000E+00
	1.8157E+04	5.3280E+03	8.0700E+02	2.9900E+02	6.0500E+02	0.0000E+00
	1.8312E+04	6.2140E+03	7.1600E+02	3.6800E+02	4.6800E+02	0.0000E+00

Example 4 shows the contents of the array parameters (*ARRAYP.INP*) file that is used as input by the Tailpipe Emissions Estimator. It contains the parameters describing the number of records and increments used in the four composite files.

Example 4. *ARRAYP.INP* file.

```
600. -8.00 1.0  2.0  4.   -300. 20
 17 20 34
```

Example 5. Sample *energy.no.out* file.

TIME	LINK	NODE	EFract1	EFract2	EFract3	EFract4	EFract5	EFract6	EFract7	EFract8
3600	1	6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000

Example 6.Sample *energy.short.out* file.

TIME	LINK	NODE	EFract1	EFract2	EFract3	EFract4	EFract5	EFract6	EFract7	EFract8
3600	1	6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Example 7. Sample *energy.med.out* file.

TIME	LINK	NODE	EFract1	EFract2	EFract3	EFract4	EFract5	EFract6	EFract7	EFract8
3600	1	6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Example 8. Sample *energy.long.out* file.

TIME	LINK	NODE	EFract1	EFract2	EFract3	EFract4	EFract5	EFract6	EFract7	EFract8
3600	1	6	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Example 9 shows a portion of the contents of the *arrayp.out* file, which is used as input to the Tailpipe Emissions Estimator. The *arrayp.out* file contains the composite vehicle emissions in 4-mph speed bins and 20-mph squared per sec bins. In this version of the Tailpipe Emissions Estimator, there are 34 power bins and 20 velocity bins. Example 9 contains data for all the velocity bins for the first two power bins (*power* = υ**acc*).

Example 9. *arrayp.out* file.

array for no grade Riverside region by power					
v	pow	hc	co	nox	fuel
2.0000	-150.0000	0.0777	0.0140	0.0008	0.4393
6.0000	-50.0000	0.0478	0.0140	0.0008	0.4393
10.0000	-30.0000	0.0264	0.0140	0.0008	0.4393
14.0000	-21.4286	0.0136	0.0140	0.0008	0.4393
18.0000	-16.6667	0.0077	0.0140	0.0007	0.4393
22.0000	-13.6364	0.0101	0.0140	0.0007	0.4393
26.0000	-11.5385	0.0096	0.0140	0.0006	0.4393
30.0000	-10.0000	0.0092	0.0140	0.0005	0.4393
34.0000	-8.8235	0.0088	0.0140	0.0004	0.4393
38.0000	-7.8947	0.0087	0.0140	0.0004	0.4393
42.0000	-7.1429	0.0086	0.0140	0.0004	0.4393
46.0000	-6.5217	0.0086	0.0140	0.0004	0.4393
50.0000	-6.0000	0.0088	0.0140	0.0007	0.4393
54.0000	-5.5556	0.0085	0.0140	0.0004	0.4393
58.0000	-5.1724	0.0106	0.0140	0.0007	0.4393
62.0000	-4.8387	0.0086	0.0140	0.0004	0.4393
66.0000	-4.5455	0.0085	0.0140	0.0004	0.4393
70.0000	-4.2857	0.0086	0.0140	0.0004	0.4393
74.0000	-4.0541	0.0087	0.0140	0.0004	0.4393
78.0000	-3.8462	0.0089	0.0140	0.0005	0.4393
2.0000	-140.0000	0.0711	0.0140	0.0008	0.4393
6.0000	-46.6667	0.0419	0.0140	0.0008	0.4393
10.0000	-28.0000	0.0220	0.0140	0.0008	0.4393
14.0000	-20.0000	0.0102	0.0140	0.0008	0.4393
18.0000	-15.5556	0.0072	0.0140	0.0007	0.4393
22.0000	-12.7273	0.0083	0.0140	0.0006	0.4393
26.0000	-10.7692	0.0092	0.0140	0.0005	0.4393
30.0000	-9.3333	0.0089	0.0140	0.0005	0.4393
34.0000	-8.2353	0.0089	0.0140	0.0005	0.4393
38.0000	-7.3684	0.0086	0.0140	0.0004	0.4393
42.0000	-6.6667	0.0086	0.0140	0.0004	0.4393
46.0000	-6.0870	0.0087	0.0140	0.0004	0.4393
50.0000	-5.6000	0.0086	0.0140	0.0004	0.4393
54.0000	-5.1852	0.0086	0.0140	0.0004	0.4393
58.0000	-4.8276	0.0088	0.0140	0.0004	0.4393
62.0000	-4.5161	0.0088	0.0140	0.0007	0.4393
66.0000	-4.2424	0.0085	0.0140	0.0004	0.4393
70.0000	-4.0000	0.0106	0.0140	0.0007	0.4393
74.0000	-3.7838	0.0085	0.0140	0.0004	0.4393
78.0000	-3.5897	0.0085	0.0140	0.0004	0.4393

<u>Example 10</u> shows a portion of the contents of an *emissions.ldv.out* file that is created by the Tailpipe Emissions Estimator. The *emissions.ldv.out* file is used as input into the Output Visualizer. This table contains the data for timestep 3600 link 1 running from node 6 as seen in the above examples.

Example 10. *emissions.ldv.out* file.

TIME	LINK	NODE	DISTANCE	LENGTH	VTT	NOX	CO	HC	FE	FLUX
3600	1	6	240.0	30.0	2.4	8776.5	2252.3	275591.6	69939.1	9625.0
3600	1	6	270.0	30.0	3.0	14090.2	1019.5	108810.8	27401.2	52347.0
3600	1	6	300.0	30.0	3.9	49170.6	1746.7	139996.8	32516.7	75231.0
3600	1	6	330.0	30.0	3.9	46140.9	1722.2	138839.1	32540.7	76294.0
3600	1	6	360.0	30.0	3.9	39328.9	1550.8	138247.0	32173.3	76335.0
3600	1	6	390.0	30.0	3.9	17635.3	1110.6	176770.5	31114.6	76648.0
3600	1	6	420.0	30.0	3.9	17102.1	1093.4	146050.2	31051.2	76748.0
3600	1	6	450.0	30.0	3.9	16028.2	1087.7	178704.3	31016.3	76654.0
3600	1	6	480.0	30.0	3.9	10638.5	1040.3	362052.7	30511.9	77609.0
3600	1	6	495.0	15.0	4.0	5249.6	529.7	186391.5	15410.0	80590.0

Example 11 shows an example of where the fields are and a sample portion of the contents of a debugging (*debug.ldv.out*) file that is created by the Tailpipe Emissions Estimator. The *debug.ldv.out* file is a debugging file used to provide immediate output for the emissions calculations. **Example 11** contains the output for calculations done on the link for the first couple of boxes. The first box will always contain calculations for the intersection.

Example 11a. Where fields are in a debugging (debug.ldv.out) file.

The sections below are repeated for each link in the network.

The following section is repeated for each segment in a particular link.

```
icx=    ICX  deltaf=  DELTAF
   COUNTS[1]    COUNTS[2]   COUNTS[3]   COUNTS[4]   COUNTS[5]  COUNTS[6]
      DEN[1]       DEN[2]      DEN[3]      DEN[4]      DEN[5]     DEN[6]
         FIJ[1]       FIJ[2]      FIJ[3]      FIJ[4]      FIJ[5]     FIJ[6]
         HIJ[1]       HIJ[2]      HIJ[3]      HIJ[4]      HIJ[5]     HIJ[6]


    VEHFLUX[1]   VEHFLUX[2]   VEHFLUX[3]   VEHFLUX[4]   VEHFLUX[5]   VEHFLUX[6]  VEHFT
       VEHD[1]      VEHD[2]      VEHD[3]      VEHD[4]      VEHD[5]      VEHD[6]  VEHDT
        VBAR         SDEVRAT      VLOWRI       VUPPRI       V2SDEV
        ITAR         IREF         SPDCL        SPDCM        SPDCH   VEHDL   VEHDM   VEHDH
    VEHFLUXL[ICX]  VEHFLUXL[ICX+1]  VEHFLUXL[ICX+2]  VEHFLUXL[ICX+3]  VEHFLUXL[ICX+4]
    VEHFLUXM[ICX]  VEHFLUXM[ICX+1]  VEHFLUXM[ICX+2]  VEHFLUXM[ICX+3]  VEHFLUXM[ICX+4]
    VEHFLUXH[ICX]  VEHFLUXH[ICX+1]  VEHFLUXH[ICX+2]  VEHFLUXH[ICX+3]  VEHFLUXH[ICX+4]
   VCUBEDL[ICX]   VCUBEDL[ICX+1]   VCUBEDL[ICX+2]   VCUBEDL[ICX+3]   VCUBEDL[ICX+4]
   VCUBEDM[ICX]   VCUBEDM[ICX+1]   VCUBEDM[ICX+2]   VCUBEDM[ICX+3]   VCUBEDM[ICX+4]
   VCUBEDH[ICX]   VCUBEDH[ICX+1]   VCUBEDH[ICX+2]   VCUBEDH[ICX+3]   VCUBEDH[ICX+4]
```

The following section is repeated for each segment in a particular link. It is outputed from the power balance routine that is called for the slowest, middle, and fastest thirds of vehicles.

```
PDL  PL   PNS   PTOTFL   PTOTMI   PTOTPP   PPNS
PDC  PCC  PNS   PTOTFL   PTOTMI   PTOTPP   PPNS
PDH  PH   PNS   PTOTFL   PTOTMI   PTOTPP   PPNS
```

The following section is repeated for each segment in a particular link

```
ICX   XNOSUL   XNOSUC   XNOSUH   COSUL   COSUC   COSUH   V2SDEV   SDEV PL PCC PH
```

Example 11b. Portion of a debugging (*debug.ldv.out*) file.

```
icx=    1  deltaf=  24.6
  42854.0   1851.4      0.0      0.0      0.0      0.0
    925.7   1851.4      0.0      0.0      0.0      0.0
       0.000       0.765       0.000       0.000       0.000       0.000
       0.124      -0.062       0.000       0.000       0.000      -0.000



    1897.644    9488.220       0.000       0.000       0.000       0.000   11385.863
     231.420     462.840       0.000       0.000       0.000       0.000     694.260
       16.40     0.46771       16.16       23.59        0.00
   2   1  0.0000  0.0000  0.0000  365.29  193.57  135.40
  3795.319     0.205 20270.137 20715.150 20701.314
  3795.281     0.205 20269.943 20714.967 20701.125
  3795.263     0.205 20270.100 20715.119 20701.289
     560648.         30.    5513990.    6297994.    6489870.
    1510915.         82.   29209592.   33279824.   33072608.
    3096138.        167.  172417632.  170984992.  170686176.
icx=    2  deltaf=  24.6
   3968.0      0.1      0.0      0.0      0.0      0.0
      0.1      0.1      0.0      0.0      0.0      0.0
       0.000       0.000       0.000       0.000       0.000       0.000
       0.000      -0.000       0.000       0.000       0.000      -0.000



       0.103       0.512       0.000       0.000       0.000       0.000       0.615
       0.013       0.025       0.000       0.000       0.000       0.000       0.038
       16.40     0.46771       16.16       23.59        0.00
   3   1  0.0182  0.3376  6.3012    0.02    0.01    0.01
     0.205 20270.137 20715.150 20701.314 20765.889
     0.205 20269.943 20714.967 20701.125 20765.699
     0.205 20270.100 20715.119 20701.289 20765.855
         30.    5513990.    6297994.    6489870.    6352844.
         82.   29209592.   33279824.   33072608.   33146338.
        167.  172417632.  170984992.  170686176.  169009648.
icx=    3  deltaf=  24.6
  16381.0   5346.0    551.0    218.0    722.0      0.0
   2452.0   5346.0    551.0    218.0    722.0      0.0
       0.000       2.209       0.228       0.090       0.298       0.000
       0.329      -0.150      -0.011       0.000      -0.024      -0.000



    5026.600   28304.348    6435.975    4022.100   17021.150       0.000   60810.180
     613.000    1336.500     137.750      54.500     180.500       0.000    2322.250
       26.19     0.91141       23.38       66.93     8753.68
   4   2  0.0235  0.4186  5.9849 1490.93  608.36  222.96
 20270.137 20715.150 20701.314 20765.889 20772.725
 20269.943 20714.967 20701.125 20765.699 20772.553
 20270.100 20715.119 20701.289 20765.855 20772.674
    5513990.    6297994.    6489870.    6352844.    6320810.
   29209592.   33279824.   33072608.   33146338.   31790422.
  172417632.  170984992.  170686176.  169009648.  169459360.
```

<u>Example 12</u> shows a portion of the contents of a debugging file (*calcsum.ldv*) that is created by the Tailpipe Emissions Estimator. The *calcsum.ldv* file is also used to provide immediate output for emissions calculations. Example 12 contains the output for calculations performed on the link seen in Example 2 and Example 3.

Example 12. Debugging (*calcsum.ldv*) file.

1.	-0.05	0.00	0.11	-0.1	0.0	1.2	-0.8	-3.0	15.2	0.003	0.012	0.00.6
0.075	0.00.8	0.559	0.	0.014	0.252	4.095	7.670	365.	194.	135.	0.000	
2.	-0.00	0.00	0.00	-0.0	0.0	0.1	-0.1	-0.2	1.2	0.003	0.012	0.006
0.075	0.008	0.559	0.	0.014	0.252	4.095	7.670	0.	0.	0.	1.333	
3.	-0.06	0.00	0.43	-0.4	4.7	2.5	-0.0	0.0	0.7	0.004	0.014	0.011
0.128	0.007	0.893	8754.	0.014	0.252	4.095	23.866	1491.	608.	223.	-0.699	
4.	-0.03	0.0	0.27	-0.4	1.3	2.3	-0.3	0.0	5.0	0.004	0.015	0.012
0.146	0.007	0.893	9762.	0.016	0.280	3.984	24.046	1453.	577.	232.	-0.009	
5.	-0.03	0.00	0.22	-0.4	-0.0	2.1	-0.3	-0.0	5.0	0.004	0.013	0.012
0.114	0.014	0.820	9976.	0.010	0.166	1.835	24.049	1433.	571.	233.	-0.015	
6.	-0.03	0.00	0.17	-0.4	-0.0	1.5	-0.5	-0.2	4.0	0.004	0.009	0.012
0.069	0.147	0.050	9677.	0.002	0.017	-0.198	23.906	1453.	578.	235.	-0.021	
7.	-0.03	0.00	0.16	-0.6	0.0	1.1	-3.1	-0.2	0.4	0.004	0.009	0.035
0.064	0.095	0.050	9503.	-0.000	-0.022	-0.123	23.795	1457.	586.	235.	-0.019	
8.	-0.03	0.00	0.15	-0.9	0.0	1.1	-2.3	-1.3	0.3	0.004	0.009	0.042
0.064	0.128	0.049	9491.	-0.000	-0.028	-0.173	23.658	1442.	584.	237.	-0.036	
9.	-0.02	0.00	0.15	-1.1	0.0	1.1	-3.8	-1.0	0.3	0.002	0.009	0.042
0.063	0.444	0.048	9121.	-0.001	-0.034	-0.243	23.419	1449.	589.	239.	-0.049	
10.	-0.03	0.00	0.16	-1.9	-2.6	1.1	-13.4	-0.0	0.0	0.003	0.009	0.088
0.061	0.863	0.037	8564.	-0.004	-0.081	-0.635	23.093	1488.	611.	245.	-0.185	
11.	-0.04	-1.08	0.18	-2.5	-3.5	1.0	-17.2	-0.2	0.2	0.004	0.008	0.154
0.054	0.881	0.019	5984.	-0.008	-0.168	-1.290	21.048	1648	724.	280.	-0.324	

Example 13 shows the Output Visualizer emissions colormaps used with the data seen in these examples to color the network's boxes. Thresholds and their colors are defined in the colormaps. See the section on Visualization for interpretation of this file.

Example 13. Output Visualization emissions colormaps.

```
5 0.0 80.0 Emissions Velocity Map 2
 5.0 3
10.0 5
20.0 9
40.0 0
80.0 8
5 0.0 130000.0 Emissions Nitrogen Oxide Map 3
  8125.0 8
 16250.0 0
 32500.0 9
 65000.0 5
130000.0 3
5 0.0 3600.0 Emissions Carbon Monixide Map 4
 225.0 8
 450.0 0
 900.0 9
1800.0 5
3600.0 3
5 0.0 200000.0 Emissions Hydrocarbons Map 5
 12500.0 8
 25000.0 0
 50000.0 9
100000.0 5
200000.0 3
5 0.0 20000.0 Emissions Fuel Economy Map 6
 1250.0 8
 2500.0 0
 5000.0 9
10000.0 5
20000.0 3
5 0.0 80000.0 Emissions Flux Map 7
 5000.0 8
10000.0 0
20000.0 9
40000.0 5
80000.0 3
```

Appendix O: Detailed Calculation Description with Reference to Examples

The objective of the Tailpipe Module is to

1) construct a continuous distribution of vehicle numbers by speed group (the f_{ij}'s and h_{ij}'s describe this),
2) estimate the fraction of vehicles engaging in intermediate, high-power, or hard-braking driving, and
3) use the three-cities power distribution to distribute the high-power vehicles into power bins (20 mph squared per second).

We also distribute the hard-braking vehicles into bins. Once we have done this, we have the elements we need to calculate emissions for vehicles with constant power.

We need to first estimate cell densities for the intersection preceding the link. The first step is to calculate v_1 and v_2 using the relationships under Section 1.4.1. Note that the counts N_{i1} are given in Example 2 for distance 270 as: 16381, 5346, 551, 218, 722, and zero. The line for distance 270 is used because there are no moving vehicles for the distances 30 through 240, since there are zero counts for all bins above zero. We obtained values of 3.23 and 3.15 for v_1 and v_2 respectively, so that we use the linear extrapolation given by the fourth relationship in Section 1.4.1. to get the estimated counts for the intersection segment denoted as N_{i0}. It is these values that appear as the second and third lines under Example 11.

The next step is to calculate the f_{i0} with the expression for f_{ij}. The resulting values are found in the fourth line of Example 11. To calculate the h_{i0}, we start with the uppermost non-zero cell and require that $d_{i0}(\Delta/2)$ be zero, so that

$$f_{40} + h_{40} \bullet \Delta / 2 = 0$$

so that h_{40} is -.026 since Δ is 24.6. We use continuity at the cell boundaries to determine the remaining gradient terms, so that

$$f_{i-10} + h_{i-10} \bullet \Delta / 2 = f_{i0} - h_{i0} \bullet \Delta / 2$$

but the use of this relationship would give h_{30} as .047, which would produce negative densities near the lower cell boundary of δ=-$\Delta/2$. Consequently, h_{30} is set to zero. The h_{00} is always set to zero to avoid negative fluxes within the zero speed bin.

Next we calculate the contribution to the *flux* for each speed bin:

$$flux_i = \int_{-\Delta/2}^{\Delta/2} \left[\Delta \bullet (i-1) + \delta v\right] \bullet \left[f_{i0} + h_{i0}\delta v\right] \bullet d\delta v$$

for speed bins 0 through 5, and the final value is the sum over all speed bins; that is, it is the total *flux*. We estimate the total density for each speed bin as

$$density = \int_{-\Delta/2}^{\Delta/2} \left[f_{i0} + h_{i0}\delta v \right] \bullet d\delta v,$$

and the total density obtained by summing over all the speed bins.

The next line gives *vbar*, *sdevrat*, *vlowri*, *vuppri*, and *v2sdev*. *vbar* is the average speed given by

$$\bar{v} = \frac{flux_t}{density_t}$$

while *sdevrat* is the ratio of the standard deviation of speed to the average speed. For this we need the second moment of speed:

$$\overline{v^2} = \frac{\sum_i \int_{-\Delta/2}^{\Delta/2} \left[\Delta \bullet (i-1) + \delta v \right]^2 \bullet \left[f_{i0} + h_{i0}\delta v \right] \bullet d\delta v}{density_t},$$

so that the standard deviation σ is given by:

$$\sigma = \sqrt{(\overline{v^2} - \bar{v} \bullet \bar{v})},$$

and *sdevrat* is:

$$\sigma_{rat} = \frac{\sigma}{\bar{v}}.$$

The next step is to find the breakpoints between slowest third (*vlowri*) and the middle third and between the fastest third (*vuppri*) and the middle third of the *flux*. We began by finding the index i_{low} such that:

$$\sum_{i=0}^{i=i_{low}-1} flux_i < flux_t / 3 < \sum_{i=0}^{i=i_{low}} flux_i,$$

and we then solve:

$$\int_{-\Delta/2}^{\delta_{low}} \left[\Delta \bullet (i-1) + \delta v \right] \bullet \left[f_{i_{low}0} + h_{i_{low}0}\delta v \right] \bullet d\delta v = flux_t / 3 - \sum_{i=0}^{i=i_{low}-1} flux_i,$$

for δ_{low}. Normally, this involves solving a cubic equation, but it sometimes becomes a quadratic equation. Once δ_{low} has been calculated, *vlowri* is calculated as:

$$vlowri = \delta_{low} \bullet \Delta \bullet (i_{low} - 1)$$

The procedure for *vuppri* is very similar except that the coefficient of $flux_t$ is two-thirds rather than one-third in the inequalities and the equation for δ_{low}. The variable *v2sdev* is given by

$$v2sdev = \bar{v}^2 \bullet (\sigma - \sigma_r)$$

The next two parameters, *itar* and *iref*, give the two segments from which the gradient will be calculated. For segments greater the one and less than the number of segments, the gradients would be calculated by differences between the segment preceding the one denoted by *icx* and the one following *icx* to give centered differences. In the case of the first segment, we use a simple forward difference as an estimate of the gradient so that *iref* is one and *itar* is two.

We now estimate the speed parameter (*sp*) for each third of the *flux*, *spdcl* for the slowest third, *spdcm* for middle third, and *spdch* for the fastest third. We need to estimate the average cube of the velocity for each third of the *flux* and for each segment in the gradient estimation. Thus we need

$$\overline{v^3} = \frac{\sum \int_{-\Delta/2}^{\delta_t} [\Delta \bullet (i-1) + \delta v]^3 \bullet [f_{ij} + h_{ij} \bullet \delta v] \bullet d\delta v}{\sum \int_{-\Delta/2}^{\delta_t} [f_{ij} + h_{ij} \bullet \delta v] \bullet d\delta v},$$

where j refers to either *itar* or *iref* and δ_t refers to the top slowest third, and the sums run over only the speed bins in the slowest third. In order to get the middle third, we can calculate a numerator that runs to the top of the second third and subtract from it the numerator above (called *vcubedl*) to the get the numerator (called *vcubedm*) for the second third. We can also calculate the numerator for the top third (*vcubedh*) by calculating a numerator that covers all of the speed bins in entirety and subtract out the numerator for the slowest two-thirds. Similar procedures are used to estimate the denominators for the middle(*vehdm*) and fastest thirds (*vehdl*). The speed parameter for the slowest one-third is

$$spdcl = \frac{\overline{v^3}_{itar} - \overline{v^3}_{iref}}{(itar - iref) \bullet 4\Delta^3},$$

where the averages are for the slowest third only. Averages for the middle and highest thirds produce *spdcm* and *spdch*, respectively.

In the output file, *itar* and *iref* are followed by *spdcl*, *spdcm*, *spdch*, *vehdl* (density for the slowest one-third), *vehdm*, and *vehdh*.

The next line gives the fluxes for the slowest third for the current cell followed by the values for the four segments. The *flux* for the slowest third is given by

$$vehfluxl = \sum_{i=0}^{i=i_{low}-1} flux_i + \int_{-\Delta/2}^{\delta_{low}} [\Delta \bullet (i_{low} - 1) + \delta v] \bullet [f_{i_{low}j} + h_{i_{low}j} \delta v] \bullet d\delta v,$$

which is merely one-third of the total *flux* if we have calculated δ_{low} correctly. The other thirds of the *flux* are calculated similarly. We could have used just the thirds of the total *flux*, but this way we have a check on the accuracy of the calculation of the breakpoints between the *flux* groups. The next two lines give *vehfluxm* for the middle third and *vehfluxh* for the fastest third, respectively. In each column, the three lines should give approximately the same value.

The last three lines give the *vcubedl, vcubedm,* and *vcubedh*, respectively for the five segments starting with *icx*. This version of the output file no longer includes several values that were calculated in older versions of the code. Code improvements made many of the previous outputs irrelevant. Once we have determined the *spdc*'s for each third, we estimate the power for an average trajectory associated with the aforementioned *spdc*'s in conjunction with *v2sdev*. We use empirical relationships that are embodied in data statements for this task. The data statements cover the three types of segments and the three speed groups. The three types of segments are: (1) freeways and arterials except for the last three segments of arterials, (2) the last three segments of arterials, and (3) freeway on-ramps. The speed groups are low, middle, and high based on *flux*. For the high power driving, we have two constants for each circumstance; dominated by *spdc* or dominated by *v2sdev*. For example, the power in the high-power driving normalized to an average trajectory for power dominated by *spdc* is

$$powa1 = p\max_{0i}(ilc) + p\max_{1i}(ilc) \bullet spdc$$

where *ilc* takes on the values 1 to 9 with 1, 2, and 3 representing the slowest one-third, medium one-third, and fastest one-third for freeways and arterials (excepting the last three segments of the arterials). *ilc* would take on the values 4, 5, and 6 for the ends of arterials, and 7, 8, and 9 for freeway on-ramps.

Similarly, for high-power driving dominated by *v2sdev* we have

$$powa2 = p\max_{2i}(ilc) + p\max_{3i}(ilc) \bullet v2sdev\,.$$

The coefficients in the preceding equations were derived for a threshold of 50 mph squared per second, so that a correction must be made for circumstances that require a higher power threshold. Specifically, for the freeway on-ramps with a threshold of 90 rather than 50, we multiply $powa2$ by the factor

$$e^{-\alpha \bullet 40}.$$

The high-power driving normalized to an average trajectory is then

$$powa = Max(powa1, powa2)$$

For hard-braking we have

$$powd1 = -(pdm_{0i}(ilc) + pdm_{1i}(ilc) \bullet spdc)$$

and

$$powd2 = -(pdm_{2i}(ilc) + pdm_{3i}(ilc) \bullet v2sdev)$$

and finally

$$powd = Max(powd1, powd2).$$

In cases where a threshold for the definition of hard-braking is –90 mph squared per second (*ef=2*) rather than –50, we multiply *powd* by a factor of

$$e^{-\beta \bullet 40}.$$

The expression for P_{ref} in Section 1.4.1 (*Algorithms*) describes how the probabilities of high-power events are estimated, except that Pow_k is used instead of its equivalent, *powa*.

However, there is a complication: we have calculated the vehicles starting with speed *vsp* and having a power associated with the three-cities power distribution curve index *ipa*; in other words we have the position, speed, and power at the start of a second of vehicle travel, but we need the position, speed, and power at the end of the second of travel to be consistent with the emission arrays. Since we assume that the power is constant over the second of travel. we have the power at the end of the second, but we still need the speed and position at the end of the second. Of course, if we have the power and the starting speed we can calculate the new speed and the new position because we can calculate the acceleration from the power, and with the acceleration, we can calculate the new speed. We have used the three-cities distribution to compute the associated accelerations: *api(icv,ivv,ipa,ialph). icv* is the coarse speed bin (7.5 m/sec), while *ivv* is fine speed bin within the coarse speed bin (8 sub-bins in a coarse speed bin). *ipa* is the aforementioned power index that we are using to approximate the three-cities distribution. *ialph* refers to the specific curves we are using for the three-cities data; remember that alpha is different for the first coarse speed bin than it is for the other speeds. We also use a different value of alpha for freeway on-ramps.

We use *vsp* and *api* to calculate *vspn*—the new speed at the end of the second. However, there is a complication in calculating the new position. We only know the starting position to within a 30-meter segment, and depending upon where the car starts in a segment, it can end up in different segments. We deal with this by assuming that the vehicles have a uniform spatial distribution within the segment and estimate the fraction *fu* the ends in the more distant segment, *icxu*, and the fraction, *fl*, that end up closest to the starting segment, *icxl*, (perhaps even in it).

dpc gives the fraction of high-power vehicles that have the three-cities power index *ipa*. Remember that we are using several points associated with the index *ipa* to approximate the three-cities curves. Essentially, the *ipa*'s correspond to equidistant points in power space along the curve of cumulative probability versus power for the high-power vehicles. The relationship between power and *ipa* is

$$P = \frac{.3 \bullet (ipa - .5)}{\alpha} + \frac{e_0}{\alpha},$$

with e_0 as the threshold used to define high-power driving (normally 40 mph squared per second). At this point, we have developed the information needed to calculate *fcont(ia,ispn,icx)* the joint frequency distribution of speed and power organized by distance down the link.

This discussion all relates to vehicles traveling with constant power, but we also have to make corrections for vehicles that are changing power, because they can have different emissions than those traveling at constant power. We assume that each car will have the same aggressiveness from one-second to the next; that is, if the car at the present time was at the 80th percentile, it will be at the 80th percentile in the preceding second. We know the speed at the end of the preceding second; it is *vsp*, so we can estimate the fraction of vehicles by segment for the preceding second using our assumption of a uniform spatial distribution within the segment. For vehicles that do not change segments during the preceding segment, they will have the same power. For vehicles that do change positions, their percentile will be the same but the power level may change because the fraction of vehicles with high-power driving may be different for the preceding speed and segment. With this approach, we can calculate the power at the preceding second and calculate the fractional power change relative to the current power. *Fcontp* is merely the fraction times *fcont*. We restrict the fraction to be between zero and one, because the emissions difference arrays are for jumps between the current power and zero power.

In order to calculate the transient effects on emissions, we need to estimate the power in the preceding second. They are two conditions that must be dealt with depending upon whether we are in the first segment or not. If we are in the first segment and the speed is less than 1 segment per second, some portion of the vehicles in the segment will have been in the segment the preceding second. We can estimate the fraction as

$$den_f = \frac{\left[\left(\frac{3}{2}den_1 - \frac{1}{2}den_2\right)v_{ref} + \left(den_1 - den_2\right)\frac{v_{ref}^2}{8\Delta}\right]}{\left(4\Delta den_1\right)},$$

with den_1 as the average vehicle density in the first segment for the speed group under consideration, den_2 as the average vehicle density in the second segment for the speed group under consideration, and v_{ref} as the speed. For the vehicles that have spent no previous time in the first segment, we assume that they have an initial power associated with near zero speed gradients and a small standard deviation in speeds. In other words, they have minimal initial power pow_{01}. The remaining vehicles in the segment can be treated with the formulation used for other segments. In this case, since the probability of a high-power event is the same in this segment as it was in the segment the vehicle was in during the preceding second (both being segment one), the preceding power is the same as the current power. Consequently, we have for the average power of the preceding second

$$pow_o = (1. - den_f)pow + den_f pow_{01}.$$

In the more general case, we have to consider how power changes along the link. We assume that a vehicle will maintain its relative aggressiveness as it moves along a link. For example, a vehicle that is in the top ten percent of all vehicles in its speed group in terms of power at one point on the link, will still be in the top ten percent at the next point in the link. Its power may change if the fraction of vehicles driving with high power or hard-braking changes, but its relative position, ordered by power, will remain the same. For a vehicle in the high-power grouping, the fraction of vehicles with higher power is:

$$p_{frac} = p_+ e^{-.3 \bullet ipa},$$

with p_+ as the fraction of vehicles in the speed group (low, middle, or high thirds) in high-power driving. The index *ipa* is the aggressiveness index; we use eighteen levels of aggressiveness to approximate the part of the three-cities power distribution that represents high-power driving. In considering the segment at the end of the previous second (actually the current position since emissions are based on the end of the second powers and speeds), the fraction of vehicles in high-power driving is p_{+p}. There are two possibilities if $p_{+p} \geq p_+$, the vehicle will have fallen within the high-power grouping at the previous second. In this situation, we estimate the previous power as

$$pow_o = \frac{.3 \bullet ipa}{\alpha} - \frac{\ln\left(\frac{p_+}{p_{+p}}\right)}{\alpha} + \frac{e_0}{\alpha},$$

with e_0 as the threshold power for high-power events. On the other hand, if $p_{+p} \lhd p_+$, the vehicle will not fall into the high-power driving in the previous second and its power will be the average power within the intermediate power group of the speed bin. With pow_o estimated, we can calculate the fractional power change from the previous second to the current second as;

$$dp_{frac} = \frac{pow - pow_o}{pow}.$$

We force dp_{frac} into the range zero to one and obtain *fcontp* as:

$$fcontp = dp_{frac} \bullet fcont.$$

For high-power driving, we only consider positive changes and we don't consider cases where we had power fractions greater than one because of slightly negative power driving in the previous second.

Appendix P: Error Codes

Emissions (ENV) codes may range between the values of 15000 and 15999.

Code	Description
15000	Not used currently.
15001	Not used currently.
15002	Not used currently.
15003	Occurs when an exception was caught within the emissions codes. Typically, this is found around calls to the Network subsystem. Exits.
15004	Occurs when an emissions module was invoked with the wrong number of command line arguments. Exits.
15005	Occurs when one of the emissions modules was unable to allocate enough storage. Exits.
15006	Occurs when one of the emissions modules was unable to read the header line from a microsimulation output file containing field names. Exits.
15007	Occurs when unable to find the link in the network link table. The data on these links are skipped.
15008	Occurs when a link is found to have an invalid street type. This could occur if velocity data was collected on street types other than those handled by the emissions modules (for instance walkways and bike paths). The data on these links are just skipped.
15009	Occurs when there is an instance where there are not continuous velocity data for a section of a link. Since emissions can only be calculated for continuous data, this link's data is skipped. The velocity data files are specified by the `EMISSIONS_LDV_VELOCITY_FILE`, `EMISSIONS_HDV_VELOCITY_FILE`, `EMISSIONS_MICROSIM_LDV_VELOCITY_FILE`, and `EMISSIONS_MICROSIM_HDV_VELOCITY_FILE` configuration file keys.
15010	Occurs when there is data in the first velocity speed bin of a box and no data in the other speed bins. Since emissions cannot be calculated for this instance, this link's data is skipped.
15011	Occurs when one of the following configuration file keys were not defined in the configuration file: `NET_DIRECTORY`, `NET_NODE_TABLE`, or `NET_LINK_TABLE`. Exits.
15012	Occurs when *EmissionsEstimator* was unable to open the postprocessed microsimulation velocity summary file. Make sure the `EMISSIONS_LDV_VELOCITY_FILE` configuration file key specifying the file has the proper filename, and the file has read permissions set correctly. Exits.
15013	Occurs when *EmissionsEstimator* was unable to open one of the four representative emissions files. Make sure the configuration file keys `EMISSIONS_COMPOSITE_INPUT_FILE`, `EMISSIONS_COMPOSITE2P_INPUT_FILE`, `EMISSIONS_COMPOSITE4P_INPUT_FILE`, and `EMISSIONS_COMPOSITE6P_INPUT_FILE` specifying the files have the proper filenames, and the file has read permissions set correctly. Exits.

Code	Description
15014	Occurs when *EmissionsEstimator* was unable to open one of the four representative emissions difference files. Make sure the configuration file keys `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE`, `EMISSIONS_COMPOSITE_DIFF2P_INPUT_FILE`, `EMISSIONS_COMPOSITE_DIFF4P_INPUT_FILE`, and `EMISSIONS_COMPOSITE_DIFF6P_INPUT_FILE` specifying the files have the proper filenames, and the file has read permissions set correctly. Exits.
15015	Occurs when *EmissionsEstimator* was unable to open the parameter file specified by the `EMISSIONS_ARRAY_PARAMETERS_FILE` configuration file key. Make sure the configuration file key has the proper filename, and the file has read permissions set correctly. Exits.
15016	Occurs when *EmissionsEstimator* was unable to open one of the four energy soak distribution files. Make sure the configuration file keys `EMISSIONS_ENR_NO_SOAK_FILE, EMISSIONS_ENR_SHORT_SOAK_FILE`, `EMISSIONS_ENR_MEDIUM_SOAK_FILE`, and `EMISSIONS_ENR_LONG_SOAK_FILE` specifying the files have the proper filenames, and the file has read permissions set correctly. Exits.
15017	Occurs when *EmissionsEstimator* was unable to open one of the four soak time ratio input files. Make sure the configuration file keys `EMISSIONS_RATIOS_SHORT_SOAK_FILE`, `EMISSIONS_RATIOS_MEDIUM_SOAK_FILE`, and `EMISSIONS_RATIOS_LONG_SOAK_FILE` specifying the files have the proper filenames, and the file has read permissions set correctly. Exits.
15018	Occurs when *EmissionsEstimator* was unable to open the LDV Tailpipe Emissions Estimator output file. Make sure the configuration file key (`EMISSIONS_LDV_OUTPUT_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15019	Occurs when *EmissionsEstimator* was unable to open the first debugging output file. Make sure the configuration file key (`EMISSIONS_DEBUG1_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15020	Occurs when *EmissionsEstimator* was unable to open the second debugging output file. Make sure the configuration file key (`EMISSIONS_DEBUG2_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15021	Occurs when the file specified by the configuration file key `EMISSIONS_ARRAY_PARAMETERS_FILE` did not have the expected number of values in it. Exits.
15022	Occurs when there was no header line in one of the representative emissions files. Make sure the configuration file keys `EMISSIONS_COMPOSITE_INPUT_FILE`, `EMISSIONS_COMPOSITE2P_INPUT_FILE`, `EMISSIONS_COMPOSITE4P_INPUT_FILE`, and `EMISSIONS_COMPOSITE6P_INPUT_FILE` specifying the files have the proper filenames. Exits.

Code	Description
15023	Occurs when there was no header line in one of the four representative emissions difference files. Make sure the configuration file keys `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE`, `EMISSIONS_COMPOSITE_DIFF2P_INPUT_FILE`, `EMISSIONS_COMPOSITE_DIFF4P_INPUT_FILE`, and `EMISSIONS_COMPOSITE_DIFF6P_INPUT_FILE` specifying the files have the proper filenames. Exits.
15024	Occurs when there were not enough data items in one of the four representative emissions files. The number of power bins and number of velocity bins that are assumed in the composite emissions files are read from the array parameters file (specified by `EMISSIONS_ARRAY_PARAMETERS_FILE`). Make sure the configuration file keys `EMISSIONS_COMPOSITE_INPUT_FILE`, `EMISSIONS_COMPOSITE2P_INPUT_FILE`, `EMISSIONS_COMPOSITE4P_INPUT_FILE`, and `EMISSIONS_COMPOSITE6P_INPUT_FILE` specifying the composite emission files have the proper filenames. Exits.
15025	Occurs when there were not enough data items in one of the four representative emissions difference files. The number of power bins and number of velocity bins that are assumed in the composite emissions files are read from the array parameters file (specified by `EMISSIONS_ARRAY_PARAMETERS_FILE`). Make sure the configuration file keys `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE`, `EMISSIONS_COMPOSITE_DIFF2P_INPUT_FILE`, `EMISSIONS_COMPOSITE_DIFF4P_INPUT_FILE`, and `EMISSIONS_COMPOSITE_DIFF6P_INPUT_FILE` specifying the composite difference files have the proper filenames. Exits.
15026	Occurs when there were not enough data items in one of the three composition ratios files. Each file should have ratios for each of the four types of emissions and each of the energy bins (eight at this time). Make sure the configuration file keys `EMISSIONS_RATIOS_SHORT_SOAK_FILE`, `EMISSIONS_RATIOS_MEDIUM_SOAK_FILE`, and `EMISSIONS_RATIOS_LONG_SOAK_FILE` specifying the ratios files have the proper filenames. Exits.
15027	Occurs when either the velocity data input file contains no data or the data in the file are of the incorrect format (for example, time specified as a floating-point number instead of an integer). Exits.
15028	Occurs when there were not enough data items in the postprocessed microsimulation light-duty vehicle velocity summary file. Make sure the configuration file key `EMISSIONS_LDV_VELOCITY_FILE` specifying the velocity file has the proper filename. Check the results of the *ConvertVELfile* program run to verify that there were no serious problems with the microsimulation velocity data file. Exits.
15029	Occurs when there was no header line in one of the four postprocessed energy summary soak files. Make sure the configuration file keys `EMISSIONS_ENR_NO_SOAK_FILE`, `EMISSIONS_ENR_SHORT_SOAK_FILE`, `EMISSIONS_ENR_MEDIUM_SOAK_FILE`, and `EMISSIONS_ENR_LONG_SOAK_FILE` specifying the energy files have the proper filenames. Exits.

Code	Description
15030	Occurs when there were not enough data items in one of the four postprocessed energy summary soak files. Each of the files should have distribution fractions for each of the energy bins (eight at this time) for each of the link/node pairs in the velocity file. Make sure the configuration file keys `EMISSIONS_ENR_NO_SOAK_FILE`, `EMISSIONS_ENR_SHORT_SOAK_FILE`, `EMISSIONS_ENR_MEDIUM_SOAK_FILE`, and `EMISSIONS_ENR_LONG_SOAK_FILE` specifying the energy files have the proper filenames. Check the results of the *ConvertENRfile* program run to verify that there were no serious problems with the microsimulation energy data files. Exits.
15031	Occurs when the four energy files are not matched properly for time, link, and node. Check the results of the *ConvertENRfile* program run to verify that there were no serious problems with the microsimulation energy data files. Exits in *ConvertENRfile*, but skipped in energy file in *EmissionsEstimator*.
15032	Occurs when *EmissionsEstimatorHDV* was unable to open the postprocessed microsimulation velocity summary file. Make sure the `EMISSIONS_HDV_VELOCITY_FILE` configuration file key specifying the file has the proper filename, and the file has read permissions set correctly. Exits.
15033	Occurs when *EmissionsEstimatorHDV* was unable to open the heavy-duty vehicle representative emissions file. Make sure the configuration file key `EMISSIONS_COMPOSITE_HDV_INPUT_FILE` specifying the file has the proper filename, and the file has read permissions set correctly. Exits.
15034	Occurs when *EmissionsEstimatorHDV* was unable to open the parameter file specified by the `EMISSIONS_HDV_ARRAY_PARAMETERS_FILE` configuration file key. Make sure that the configuration file key has the proper filename, and the file has read permissions set correctly. Exits.
15035	Occurs when *EmissionsEstimatorHDV* was unable to open the HDV Tailpipe Emissions Estimator output file. Make sure the configuration file key (`EMISSIONS_HDV_OUTPUT_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15036	Occurs when *EmissionsEstimatorHDV* was unable to open the first debugging output file. Make sure the configuration file key (`EMISSIONS_DEBUG1_HDV_FILE`) specifying the file has the proper filename, and the directory is writable.
15037	Occurs when *EmissionsEstimatorHDV* was unable to open the second debugging output file. Make sure the configuration file key (`EMISSIONS_DEBUG2_HDV_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15038	Occurs when the file specified by the configuration file key `EMISSIONS_HDV_ARRAY_PARAMETERS_FILE` did not have the expected number of values in it. Exits.
15039	Occurs when there were not enough data items in the heavy-duty vehicle representative emissions file. The number of speed bins that are assumed in the composite emissions file is twenty. Make sure the configuration file key `EMISSIONS_COMPOSITE_HDV_INPUT_FILE` specifying the composite emission file has the proper filename. Exits.
15040	Occurs when the heavy-duty vehicle representative emissions file was missing the two header lines in between data items in the file specified by the `EMISSIONS_COMPOSITE_HDV_INPUT_FILE` configuration file key. Exits.

Code	Description
15041	Occurs when there were not enough data items in the heavy-duty vehicle representative emissions file. The number of power bins and number of velocity bins that are assumed in the composite emissions files are read from the array parameters file (specified by `EMISSIONS_HDV_ARRAY_PARAMETERS_FILE`). Make sure the configuration file key `EMISSIONS_COMPOSITE_HDV_INPUT_FILE` specifying the composite emission file has the proper filename. Exits.
15042	Occurs when there were not enough data items in the postprocessed microsimulation heavy-duty vehicle velocity summary file. Make sure the configuration file key `EMISSIONS_HDV_VELOCITY_FILE` specifying the velocity file has the proper filename. Check the results of the *ConvertVELfile* program run to verify that there were no serious problems with the microsimulation velocity data file. Exits.
15043	Occurs when the timestep is not the assumed time. In this version of the software, the Tailpipe Emission Estimator expect a timestep of 3600 seconds (one hour). That particular velocity file is not processed. The timestep is specified by the `OUT_SUMMARY_TIME_STEP_DEFAULT` or the `OUT_SUMMARY_TIME_STEP_n` configuration file key.
15044	Occurs when the microsimulation velocity summary file does not have the assumed sample time of one second. That particular velocity file is not processed. The sample time is specified by the `OUT_SUMMARY_SAMPLE_TIME_DEFAULT` or the `OUT_SUMMARY_SAMPLE_TIME_n` configuration file key.
15045	Occurs when the microsimulation velocity summary file does not have the assumed cell length of 7.5 meters. That particular velocity file is not processed. The cell length is specified by the `CA_CELL_LENGTH` configuration file key.
15046	Occurs when the microsimulation velocity summary file does not have the assumed box length of 30 meters. That particular velocity file is not processed. The box length is specified by the `OUT_SUMMARY_BOX_LENGTH_DEFAULT` or the `OUT_SUMMARY_BOX_LENGTH_n` configuration file key.
15047	Occurs when the file specified by the configuration file key `EMISSIONS_MICROSIM_LDV_VELOCITY_FILE` does not contain data of vehicle type `AUTO`, or the file specified by the configuration file key `EMISSIONS_MICROSIM_HDV_VELOCITY_FILE` does not contain data of vehicle type `TRUCK`. That particular velocity file is not processed.
15048	Occurs when the microsimulation velocity summary file does not have the assumed max velocity of 37.5 meters. That particular velocity file is not processed. The max velocity is specified by the `OUT_SUMMARY_VELOCITY_MAX_DEFAULT` or the `OUT_SUMMARY_VELOCITY_MAX_n` configuration file key.
15049	Occurs when the microsimulation velocity summary file does not have the assumed number of bins of six. That particular velocity file is not processed. The number of bins is specified by the `OUT_SUMMARY_VELOCITY_BINS_DEFAULT` or the `OUT_SUMMARY_VELOCITY_BINS_n` configuration file key.
15050	Occurs when *ConvertVELfile* was unable to open the microsimulation velocity summary file. That particular velocity file is not processed. Make sure the configuration file keys (`EMISSIONS_MICROSIM_LDV_VELOCITY_FILE` and `EMISSIONS_MICROSIM_HDV_VELOCITY_FILE`) specifying the files have the proper filenames, and the files have read permissions set correctly.

Code	Description
15051	Occurs when *ConvertVELfile* was unable to open one of the velocity output files. That particular velocity file is not processed. Make sure the configuration file keys (EMISSIONS_LDV_VELOCITY_FILE and EMISSIONS_HDV_VELOCITY_FILE) specifying the files have the proper filenames, and the directories are writable.
15052	Occurs when the emissions module has determined that there is metadata in the microsimulation velocity summary file it is attempting to process, but an error occurred when it tried to read it. That particular velocity file is not processed.
15053	Occurs when a problem occurred with verifying the metadata in the microsimulation velocity summary file. That particular velocity file is not processed.
15054	Occurs when a field was missing out of the microsimulation velocity summary file. That particular velocity file is not processed. Necessary fields are TIME, LINK, NODE, DISTANCE, COUNT0, COUNT1, COUNT2, COUNT3, COUNT4, and COUNT5.
15055	Occurs when a microsimulation velocity summary file had a header but no data. That particular velocity file is not processed.
15056	Occurs when both of the light-duty and heavy-duty velocity input files have similar problems. This can occur with both unable to open input or output files, metadata not as assumed, missing header files, etc. Exits.
15057	Occurs when a microsimulation energy summary file had a header but no data.
15058	Occurs when the energy files have different timesteps (specified by OUT_SUMMARY_TIME_STEP_DEFAULT or OUT_SUMMARY_TIME_STEP_n). The energy files must all have the same timestep for their usage to be valid. Exits.
15059	Occurs when one or more of the energy files is not of the assumed soak time. Each of the four microsimulation energy summary files must be of a particular soak time. For example, for the NEGLIGIBLE file, this is specified in the configuration file with the following configuration file keys/values: EMISSIONS_MICROSIM_ENR_NO_SOAK_FILE specifies the microsimulation energy output file created by the following specification. Filename is specified by OUT_SUMMARY_NAME_n with the OUT_DIRECTORY prefix and a *.enr* extension, OUT_SUMMARY_TYPE_n ENERGY and OUT_SUMMARY_ENERGY_SOAK_n NEGLIGIBLE. EMISSIONS_MICROSIM_ENR_SHORT_SOAK_FILE comes from the OUT_SUMMARY_ENERGY_SOAK_n configuration file key set to SHORT specification. EMISSIONS_MICROSIM_ENR_MEDIUM_SOAK_FILE comes from the OUT_SUMMARY_ENERGY_SOAK_n configuration file key set to MEDIUM specification. EMISSIONS_MICROSIM_ENR_LONG_SOAK_FILE comes from the OUT_SUMMARY_ENERGY_SOAK_n configuration file key set to LONG specification. Each of these four soak files must have a unique OUT_SUMMARY_NAME_n. Exits.
15060	Occurs when all energy files have the same max energy (specified by OUT_SUMMARY_ENERGY_MAX_DEFAULT or OUT_SUMMARY_ENERGY_MAX_n) but not the assumed energy. The NEGLIGIBLE file can be 0.0, but all others must be set to 105. Exits.

Code	Description
15061	Occurs when the energy files have different max energies defined (specified by `OUT_SUMMARY_ENERGY_MAX_DEFAULT` or `OUT_SUMMARY_ENERGY_MAX_n`). The `NEGLIGIBLE` file can be set to 0.0, but all others must be set to 105. Exits.
15062	Occurs when the energy files all have the same number of bins but not the assumed number. The energy files must either all have the same assumed number of energy bins of eight; or the `NEGLIGIBLE` file may have one bin, and the other three must have eight bins each. The number of bins is specified by the configuration file key `OUT_SUMMARY_ENERGY_BINS_DEFAULT` or `OUT_SUMMARY_ENERGY_BINS_n`. Exits.
15063	Occurs when the energy files have different numbers of bins (specified by `OUT_SUMMARY_ENERGY_BINS_DEFAULT` or `OUT_SUMMARY_ENERGY_BINS_n`). The energy files must either all have the same assumed number of energy bins of eight; or the `NEGLIGIBLE` file may have one bin, and the other three must have eight bins each. Exits.
15064	Occurs when all of the energy files have the same short soak time (specified by `CA_SHORT_SOAK_TIME`), but not the assumed time. The energy files must all have the same assumed short soak time of 600 seconds (10 minutes). Exits.
15065	Occurs when the energy files have different short soak times (specified by `CA_SHORT_SOAK_TIME`). The energy files must all have the same assumed short soak time of 600 seconds (10 minutes). Exits.
15066	Occurs when all of the energy files have the same medium soak time (specified by `CA_MEDIUM_SOAK_TIME`), but not the assumed time. The energy files must all have the same assumed medium soak time of 1800 seconds (30 minutes). Exits.
15067	Occurs when the energy files have different medium soak times (specified by `CA_MEDIUM_SOAK_TIME`). The energy files must all have the same assumed medium soak time of 1800 seconds (30 minutes). Exits.
15068	Occurs when all of the energy files have the same long soak time (specified by `CA_LONG_SOAK_TIME`), but not the assumed time. The energy files must all have the same assumed long soak time of 9000 seconds (2.5 hours). Exits.
15069	Occurs when the energy files have different long soak times (specified by `CA_LONG_SOAK_TIME`). The energy files must all have the same assumed long soak time of 9000 seconds (2.5 hours). Exits.
15070	Occurs when the energy files have different date and time stamps. Date and time stamps must all match to make sure using energy files from the same microsimulation run. Exits.
15071	Occurs when some of the energy files have metadata, but not all. Either all files need to have the data or none of them. Exits.
15072	Occurs when *ConvertENRfile* was unable to open one of the microsimulation summary energy files. Make sure the configuration file keys specifying the files have the proper filenames: `EMISSIONS_MICROSIM_ENR_NO_SOAK_FILE`, `EMISSIONS_MICROSIM_ENR_SHORT_SOAK_FILE`, `EMISSIONS_MICROSIM_ENR_MEDIUM_SOAK_FILE`, and `EMISSIONS_MICROSIM_ENR_LONG_SOAK_FILE`, and the files have read permissions set correctly. Exits.

Code	Description
15073	Occurs when *ConvertENRfile* was unable to open one of the energy output files. Make sure the configuration file keys specifying the files have the proper filenames, and the directory(es) are writable: `EMISSIONS_ENR_NO_SOAK_FILE`, `EMISSIONS_ENR_SHORT_SOAK_FILE`, `EMISSIONS_ENR_SHORT_SOAK_FILE`, and `EMISSIONS_ENR_LONG_SOAK_FILE`. Exits.
15074	Occurs when a field was missing out of a microsimulation energy summary file. Necessary fields are `TIME, LINK, NODE, ENERGY0, ENERGY1, ENERGY2, ENERGY3, ENERGY4, ENERGY5, ENERGY6`, and `ENERGY7`. The `NEGLIGIBLE` file may have the fields `TIME, LINK, NODE, ENERGY0`. Exits.
15075	Occurs when *ConvertTRVfile* finds a link with a third node ID associated with it. Only two end nodes can be associated with a link. The line in the traveler event file is skipped. Exits.
15076	Occurs when *ConvertTRVfile* was unable to open the microsimulation traveler event file. Make sure the configuration file key (`EMISSIONS_MICROSIM_TRAVELER_FILE`) specifying the file has the proper filename, and the file has read permissions set correctly. Exits.
15077	Occurs when a field was missing out of the microsimulation traveler event file. Necessary fields are `TIME, VEHICLE, LOCATION, STATUS`, and `VEHTYP` for creation of the vehicle parking location/time file. Exits.
15078	Occurs when *ConvertTRVfile* was unable to open the vehicle parking location/time output file. Make sure the configuration file key (`EMISSIONS_PA_OUTPUT_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15079	Occurs when a field was missing out of the microsimulation traveler event file. Necessary fields are `TIME, LINK, NODE, STATUS, VEHTYPE`, and `VSUBTYPE` for creation of the vehicle subtype count file. Exits.
15080	Occurs when the `EMISSIONS_NUMBER_VSUBTYPES` configuration file key has a value larger than the allowed maximum subtypes. At this time, 50 subtypes are allowed. Exits.
15081	Occurs when *ConvertTRVfile* was unable to open the vehicle subtype count output file. Make sure the configuration file key (`EMISSIONS_SUBTYPE_OUTPUT_FILE`) specifying the file has the proper filename, and the directory is writable. Exits.
15082	Occurs when the microsimulation time summary file had a header but no data. Exits, not used at this time.
15083	Occurs when *ConvertTIMfile* was unable to open the microsimulation summary time file. Make sure the configuration file key specifying the file (`EMISSIONS_MICROSIM_TIME_FILE`) has the proper filename, and the file has read permissions set correctly. Exits, not used at this time.
15084	Occurs when *ConvertTIMfile* was unable to open the time output file. Make sure the configuration file key specifying the file (`EMISSIONS_TIME_FILE`) has the proper filename, and the directory is writable. Exits, not used at this time.
15085	Occurs when a field was missing out of the microsimulation time summary file. Necessary fields are `TIME, LINK, NODE, SUM, COUNT`, and `VCOUNT`. Exits, not used at this time.
15086	Occurs when *CreateComposites* was unable to open the vehicle type distribution file. The `EMISSIONS_VEHICLE_TYPE_DISTRIBUTION` configuration file key specifies this file, and the file has read permissions set correctly. Exits.

Code	Description
15087	Occurs when the file specified by the configuration file key `EMISSIONS_VEHICLE_TYPE_DISTRIBUTION` did not have the expected number of values in it. Exits.
15088	Occurs when *CreateComposites* was unable to open the representative emissions file for all vehicle subtypes. Make sure the `EMISSIONS_COMPOSITE_INPUT_FILE` configuration file key specifies the proper filename, and the file has read permissions set correctly. Exits.
15089	Occurs when *CreateComposites* was unable to open the output file that is supposed to contain the representative emissions for one combined vehicle type. Make sure the `EMISSIONS_COMPOSITE_INPUT_FILE` configuration file key contains the proper filename, and the directory has write permissions. Exits.
15090	Occurs when the file specified by the configuration file key `EMISSIONS_COMPOSITE_TYPE_INPUT_FILE` did not have the expected number of values in it. Exits.
15091	Occurs when *CreateComposites* was unable to open the representative emissions difference file for all vehicle subtypes. Make sure the `EMISSIONS_COMPOSITE_TYPE_DIFF_INPUT_FILE` configuration file key contains the proper filename, and the file has read permissions set correctly. Exits.
15092	Occurs when *CreateComposites* was unable to open the output file that is supposed to contain the representative emission differences for one combined vehicle type. Make sure the `EMISSIONS_COMPOSITE_DIFF_INPUT_FILE` configuration file key contains the proper filename, and the file has read permissions set correctly. Exits.
15093	Occurs when the file specified by the configuration file key `EMISSIONS_COMPOSITE_TYPE_DIFF_INPUT_FILE` did not have the expected number of values in it.
15094	Occurs when *EvaporativeEstimator* was unable to open the emissions coefficients input file. Make sure the `EMISSIONS_EVAP_COEF_FILE` configuration file key specifying the file has the proper filename, and the file has read permissions set correctly.
15095	Occurs when *EvaporativeEstimator* was unable to open the parking location input file. Make sure the `EMISSIONS_PA_OUTPUT_FILE` configuration file key specifying the file has the proper filename, and the file has read permissions set correctly.
15096	Occurs when *EvaporativeEstimator* was unable to open the microsimulation velocity summary input file. Make sure the `EMISSIONS_MICROSIM_LDV_VELOCITY_FILE` configuration file key has the proper filename, and the file has read permissions set correctly.
15097	Occurs when *EvaporativeEstimator* was unable to open the Stationary losses emissions output file. Make sure the `EMISSIONS_EVAP_STATIONARY_OUTFILENAME` configuration file key specifying the file has the proper filename, and the directory is writable.
15098	Occurs when *EvaporativeEstimator* was unable to open the Operating losses emissions output file. Make sure the `EMISSIONS_EVAP_OPERATING_OUTFILENAME` specifying the file has the proper filename, and the directory is writable.
15099	Occurs when the *EvaporativeEstimator* was unable to open the city-specific data input file. Make sure the `EMISSIONS_EVAP_CITY_FILE` configuration file key specifying the file has the proper filename, and the file has read permissions set correctly.

Code	Description
15100	Occurs when *EvaporativeEstimator* was unable to open the debug output file. Make sure the `EMISSIONS_EVAP_DEBUG_FILENAME` configuration file key specifying the file has the proper filename, and the directory is writable.
15101	Occurs when there was no header line in the parking location input file. Make sure the configuration file key `EMISSIONS_PA_OUTPUT_FILE` specifying the file has the proper filename.
15102	Occurs when there was no header line in the city-specific input file. Make sure the configuration file key `EMISSIONS_EVAP_CITY_FILE` specifying the file has the proper filename.
15103	Occurs when the file specified by the configuration file key `EMISSIONS_PA_OUTPUT_FILE` did not have the expected number of values in it.
15104	Occurs when the file specified by the configuration file key `EMISSIONS_PA_OUTPUT_FILE` did not have the expected number of values in it.
15105	Occurs when the file specified by the configuration file key `EMISSIONS_EVAP_CITY_FILE` did not have the expected number of values in it..
15106	Occurs when the file specified by the configuration file key `EMISSIONS_EVAP_CITY_FILE` has a value for number of years of vehicle age distribution that is greater than the maximum years of age distribution allowed in the software (50 years).
15107	Occurs when the file specified by the configuration file key `EMISSIONS_PA_OUTPUT_FILE` either has no data in it or the data that is in it has simulation times that are out of range of the simulation start and end times specified by the configuration file keys `CA_SIM_START_SECOND`, `CA_SIM_START_MINUTE`, `CA_SIM_START_HOUR`, and `CA_SIM_STEPS`.
15108	Occurs when the *EvaporativeEstimator* comes across an unknown emissions type in one of the following functions: **`askLeaker`**, **`getHotSoak`**, or **`getPartialDiurnal`**. That particular emission type is not calculated for the current vehicle type.
15109	Occurs when the *EvaporativeEstimator* comes across an hour that is out of the normal bounds (0 to 23) in the **`getTemperature`** function.
15110	Occurs when Cumulative Model Year Distribution adds up to greater than 100% when initializing the `UserDefined` variables.
15111	Occurs in **`GetParkingInfo()`** when a parking ID is not found to be in the network. The emissions data for this parking location is not outputted.
15112	Occurs when getting a parking location's offset from the intersection, and the offset is larger than the length of the link. The emissions data for this parking location is not outputted.
15113	Occurs when the *EvaporativeEstimator* comes across a read in simulation time that is lower than or higher than the simulation start and end times defined by the configuration file keys `CA_SIM_START_SECOND`, `CA_SIM_START_MINUTE`, `CA_SIM_START_HOUR`, and `CA_SIM_STEPS`.
15114	Occurs when the *EvaporativeEstimator* encounters a parking location ID of zero in the **`Emissions:setHC`** function.

Chapter Seven: Index